UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di Scienze Statistiche
Corso di Dottorato di Ricerca in Scienze Statistiche
Ciclo XXXIV

Some developments on variational approximation methods for Bayesian inference

Coordinatore del Corso: Prof. Nicola Sartori

Supervisore: Prof. Mauro Bernardi

Co-supervisore: Dott. Luca Maestrini

Dottorando: Emanuele Degani

January 12^{th}, 2022

Abstract

One of the most common and prolific aspects of modern Bayesian statistical research concerns the determination of posterior distributions. Over the last decades, Monte Carlo sampling techniques based on Markov chains have represented the primary strategy to obtain posterior distributions when their explicit form is unavailable. Nevertheless, some limitations regarding their usage have emerged with the advent of the so-called *big data*, including the impossibility of scaling to large amounts of data and the non-rare difficulty of achieving satisfactory convergence.

Variational approximations represent a possible alternative that may overcome the issues of Markov chain Monte Carlo. These methods are based on approximating the posterior distribution with a more tractable statistical distribution. The approximating distribution must be identified within a family of statistical distributions wisely chosen to be computationally manageable and sufficiently similar to the true and unknown posterior distribution. Once the approximation is obtained, this can be used to carry out inference on the model parameters. If the approximation is reliable, the inferential conclusions will be similar to those obtainable from the true posterior distribution.

This PhD thesis studies new variational approximation strategies for Bayesian inference and prediction concerning prominent statistical models and challenging data types. The first chapter provides a sufficiently exhaustive overview of the existing literature on variational approximations. The second chapter investigates alternatives to common variational approximation approaches based on the Kullback-Leibler divergence for fitting univariate, linear and generalized linear models. The third chapter develops a computationally efficient and accurate approximation strategy for selecting fixed effects in multilevel linear models. The fourth chapter proposes posterior distribution approximations for Gaussian process regression models having nonstationary covariance functions.

Sommario

Uno degli aspetti più noti e prolifici nella ricerca statistica moderna di stampo Bayesiano riguarda la determinazione delle distribuzioni a posteriori. Negli ultimi decenni, le tecniche di campionamento Monte Carlo basate su catene di Markov hanno rappresentato la strategia principale per ottenere distribuzioni a posteriori quando la loro forma esplicita non è disponibile. Ciononostante, alcuni limiti riguardanti il loro utilizzo sono emersi con l'avvento dei cosiddetti *big data*, tra i quali l'impossibilità di essere scalate su grandi quantità di dati e la non-rara difficoltà a raggiungere una soddisfacente convergenza.

I metodi di approssimazione variazionale rappresentano una possibile alternativa che potrebbe superare i problemi legati al campionamento Monte Carlo basato su catene di Markov. Questi metodi si basano sull'approssimare la distribuzione a posteriori con una distribuzione statistica più facilmente trattabile. La distribuzione approssimante va individuata all'interno di una famiglia di possibili distribuzioni statistiche, sapientemente scelta affinchè sia computazionalmente maneggevole e sufficientemente simile alla vera ed ignota distribuzione a posteriori. Una volta determinata l'approssimazione, questa può essere utilizzata per svolgere l'inferenza sui parametri del modello. Se l'approssimazione è fedele, le conclusioni inferenziali ottenute saranno simili a quelle ottenibili con la vera distribuzione a posteriori.

Questa tesi di dottorato studia nuove strategie di approssimazione variazionale per l'inferenza e la previsione di tipo Bayesiano, riguardanti importanti modelli statistici e stimolanti tipologie di dati. Il primo capitolo svolge una panoramica sufficientemente esaustiva sulla letteratura esistente riguardante le approssimazioni variazionali. Il secondo indaga alternative ai più comuni approcci di approssimazione variazionale basati sulla divergenza di Kullback e Leibler per l'adattamento di modelli univariati, lineari e lineari generalizzati. Il terzo capitolo sviluppa una strategia di approssimazione computazionalmente efficiente ed accurata per la selezione di effetti fissi in modelli lineari multilivello. Il quarto capitolo propone approssimazioni della distribuzione a posteriori per modelli di regressione basati su processi Gaussiani aventi funzioni di covarianza non-stazionarie.

To Giulia,
my exact approximation

Acknowledgements

It is well known that any PhD experience is, in most cases, punctuated by moments of discouragement, weeks of research work that never seem to have an outlet and intuitions that, after a few weeks, turn out to be wrong. My experience is not exempt from any of these events. Despite being part of a path that presupposes them as inevitable moments of personal and professional growth, I am forced to thank some people without whom facing all of this could have proved even more difficult than it should.

The most dutiful – and certainly not only formal – thanks go to the coordinatore of the PhD course, Professor Nicola Sartori, for having always made himself available to ensure that the PhD program was up to our needs, especially during the pandemic. I also want to thank many of the professors I met during my last eight years in Santa Caterina, who contributed to making me passionate about statistics, transmitting competence and passion.

Mauro, you have proved to be a patient supervisor, and you have always been available to *bend over backward* to help me over the years. Luca, working with you even with several time zones away has been a truly enriching experience, and I owe you a lot. Sincere thanks go to Dorota Toczydłowska and Matias Quiroz for allowing me to work with them. A thank you that I inevitably extend to Professor Matt Wand, the *gravity pole* around which all the people of the Sydney cluster with whom I have worked pertain. I will always regret that I had to give up my visiting programme due to the imminent escalation of the pandemic and that I never got to know each of you in person.

I thank in advance those who will be called to read, correct and judge the following pages. I hope they do not bore you more than they could. I blame myself for every unclear concept, error, inconsistency on the notation and, above all, for not containing myself with words.

I also thank my PhD cycle mates: Anam, Anna, Dung, Jacopo (x2), Laura, Lorenzo, Mattia and Silvia. Each of you has given me so much, and I hope to have left a positive memory of me as well. I wish each of you the best for the future, whatever path you decide to take, because you deserve it.

I cannot forget all the friends and colleagues *belonging to the academia* who in recent years have been able to give me honest and valuable advice: Meme, Riccardo, Pietro, Cristian, Alessandro, Tommy, Andrea and Davide, among all.

The most intimate and personal thanks are those I always find myself making every time I write a thesis. It is true, this is the most important and the one that took the most effort from me; therefore, you deserve a few more words. To my parents for always supporting me, for giving me *the foundations to build heights*, for encouraging and helping me whenever I needed. I know how much of all this has cost you, and I will be eternally grateful. Matilde and all my relatives, all of this naturally extends to you. Ai miei nonni e nonne il ringraziamento lo faccio in italiano: avete sempre creduto in me in modo incondizionato, nonostante non sia mai riuscito a spiegarvi in parole semplici cosa abbia realmente combinato durante i miei ultimi tre anni in università. Grazie. To my hometown friends: you are the most precious baggage that I have brought with me while away from home.

Giulia, I dedicated this PhD thesis to you because if it is not composed of empty pages, it is above all your merit. You are the constant of my everyday life and the best promise I have for the future.

Brescia, January 12^{th}, 2022

The Greek Alphabet

Letter (*lowercase*)	Letter (*uppercase*)	English name	Greek name
α	A	alpha	άλφα
β	B	beta	βήτα
γ	Γ	gamma	γάμμα
δ	Δ	delta	δέλτα
ϵ	E	epsilon	έψιλον
ζ	Z	zeta	ζήτα
η	H	eta	ήτα
θ	Θ	theta	θήτα
ι	I	iota	ιώτα
κ	K	kappa	κάππα
λ	Λ	la(m)bda	λά(μ)βδα
μ	M	mu	μυ
ν	N	nu	νυ
ξ	Ξ	xi	ξι
o	O	omicron	όμικρον
π	Π	pi	πι
ρ	P	rho	ρώ
σ	Σ	sigma	σίγμα
τ	T	tau	ταυ
υ	Υ	upsilon	ύψιλον
ϕ	Φ	phi	φι
χ	X	chi	χι
ψ	Ψ	psi	ψι
ω	Ω	omega	ωμέγα

Source: `https://en.wikipedia.org/wiki/Greek_alphabet`.

Contents

List of Figures

List of Tables

List of Algorithms

Notational Conventions

Throughout this PhD thesis, lower-case Roman and Greek letters denote scalars. Lower-case Roman and Greek letters in boldface denote vectors. Unless specified otherwise, they are always assumed to be column vectors. Upper-case Roman and Greek letters in boldface denote matrices, whose entries are indicated with double-index subscripts. Blackboard-bold Roman letters indicate numeric sets.

We now list and clarify notational conventions adopted in this PhD thesis:

$\mathbb{N}$ (and $\mathbb{N}_+$)	The set of natural (and positive) numbers.
$\mathbb{R}$ (and $\mathbb{R}_+$)	The set of real (and positive) numbers.
$\mathbb{R}^d$	The set of real vectors of dimension d, for $d \in \mathbb{N}$.
$\mathbb{R}^{d \times d'}$	The set of real matrices of dimension $d \times d'$, for $d, d' \in \mathbb{N}$.
$\mathbb{S}_+^d$	The set of real symmetric and positive definite matrices of dimension $d \times d$, for $d \in \mathbb{N}$.
$[\boldsymbol{a}]_i$ or a_i	The ith element of vector $\boldsymbol{a} \in \mathbb{R}^d$, for $1 \leq i \leq d$.
$[\boldsymbol{A}]_{ij}$ or A_{ij}	The (i,j)th cell of matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d'}$, for $1 \leq i \leq d$ and $1 \leq j \leq d'$.
$\boldsymbol{a}_{-i}$	The vector of dimension $(d-1) \times 1$ obtained removing the ith element of $\boldsymbol{a} \in \mathbb{R}^d$, for $d \in \mathbb{N}$.
$(\cdot)^T$	The transpose operator, such that for $\boldsymbol{a} \in \mathbb{R}^d$, $\boldsymbol{a}^T$ is a $1 \times d$ row vector and for $\boldsymbol{A} \in \mathbb{R}^{d \times d'}$, $\boldsymbol{A}^T$ is a $d' \times d$ matrix with $(\boldsymbol{A})_{ij}^T = \boldsymbol{A}_{ji}$.
$(\cdot, \ldots, \cdot)$	The concatenate operator, working horizontally.
$\boldsymbol{0}$ and $\boldsymbol{O}$	A vector and a matrix full of zeros, respectively.
$\boldsymbol{1}$ and $\boldsymbol{U}$	A vector and a matrix full of ones, respectively.
$\boldsymbol{I}$	The identity matrix.
$\text{diag}(\cdot)$	For a vector $\boldsymbol{a} \in \mathbb{R}^d$, $\text{diag}(\boldsymbol{a})$ creates the $d \times d$ diagonal matrix with all the elements of $\boldsymbol{a}$ placed on its main diagonal.

$\text{diagonal}(\cdot)$ For a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $\text{diagonal}(\boldsymbol{A})$ creates the $d \times 1$ column vector composed by the main diagonal of $\boldsymbol{A}$.

$\text{stack}\,(\cdot)$ For a sequence of m matrices $\boldsymbol{A}_1 \in \mathbb{R}^{d_1 \times d}, \boldsymbol{A}_2 \in \mathbb{R}^{d_2 \times d}, \ldots, \boldsymbol{A}_m \in \mathbb{R}^{d_m \times d}$, $\text{stack}_{1 \leq i \leq m}\,(\boldsymbol{A}_i)$ creates the $(d_1 + d_2 + \ldots + d_m) \times d$ matrix obtained stacking all the matrices underneath each other vertically.

$\text{blockdiag}\,(\cdot)$ For a sequence of m matrices $\boldsymbol{A}_1 \in \mathbb{R}^{d_1 \times d'_1}, \boldsymbol{A}_2 \in \mathbb{R}^{d_2 \times d'_2}, \ldots, \boldsymbol{A}_m \in \mathbb{R}^{d_m \times d'_m}$, $\text{blockdiag}_{1 \leq i \leq m}\,(\boldsymbol{A}_i)$ creates the $(d_1 + d_2 + \ldots + d_m) \times (d'_1 + d'_2 + \ldots + d'_m)$ block-diagonal matrix obtained placing all matrices in its main diagonal.

$\text{vec}(\cdot)$ For a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d'}$, $\text{vec}(\boldsymbol{A})$ returns the vector of dimension $dd' \times 1$ obtained stacking the columns of $\boldsymbol{A}$ underneath each other in order from left to right. $\text{vec}^{-1}(\cdot)$ is the associated inverse operator: for a vector $\boldsymbol{a} \in \mathrm{R}^{dd'}$, $\text{vec}^{-1}_{d \times d'}(\boldsymbol{a})$ returns the $d \times d'$ matrix such that $\text{vec}(\text{vec}^{-1}_{d \times d'}(\boldsymbol{A})) = \boldsymbol{A}$. The subscript must be coherent with the vector dimension, and can be omitted if and only if $d = d'$.

$\text{vech}(\cdot)$ For a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $\text{vech}(\boldsymbol{A})$ returns the vector of dimension $d(d+1)/2 \times 1$ obtained *half-vectorizing* $\boldsymbol{A}$, that is, stacking the lower-diagonal columns of $\boldsymbol{A}$ (the diagonal elements included) underneath each other in order from left to right. $\text{vech}^{-1}(\cdot)$ is the associated inverse operator: for a vector $\boldsymbol{a} \in \mathbb{R}^{d(d+1)/2}$, $\text{vech}^{-1}(\boldsymbol{a})$ returns the $d \times d$ symmetric matrix such that $\text{vech}(\text{vech}^{-1}(\boldsymbol{A})) = \boldsymbol{A}$.

$\dim(\cdot)$ For a generic space $\mathbb{A}$, $\dim(\mathbb{A})$ returns its dimension.

$\text{tr}(\cdot)$ For a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $\text{tr}(\boldsymbol{A})$ returns its trace $\sum_{i=1}^{d} A_{ii}$.

$|\cdot|$ For a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $|\boldsymbol{A}|$ returns its determinant.

$(\cdot)^{-1}$ For a non-singular matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, $\boldsymbol{A}^{-1}$ returns its inverse.

$\otimes$ For two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p \times q}$, $\boldsymbol{A} \otimes \boldsymbol{B}$ returns a $mp \times nq$ matrix obtained by *Kronecker product*.

$\|\cdot\|$ For a matrix $\boldsymbol{a} \in \mathbb{R}^{d}$, $\|\boldsymbol{a}\|$ returns its Euclidean norm $(\sum_{i=1}^{d} a_i^2)^{1/2}$.

$+, -, \odot, /$ For two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, $\boldsymbol{a}\{+, -, \odot, /\}\boldsymbol{b}$ returns the $d \times 1$ vector obtained {adding, subtracting, multiplying, dividing} $\boldsymbol{a}$ and $\boldsymbol{b}$ elementwise. For two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d \times d'}$, $\boldsymbol{A}\{+, -, \odot, /\}\boldsymbol{B}$ returns the $d \times d'$ matrix obtained {adding, subtracting, multiplying, dividing} $\boldsymbol{A}$ and $\boldsymbol{B}$ elementwise.

$f(\cdot)$ For a vector $\boldsymbol{a} \in \mathbb{R}^d$, $f(\boldsymbol{a})$ returns the $d \times 1$ vector obtained computing f elementwise. Similarly, for a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d'}$, $f(\boldsymbol{A})$ returns the $d \times d'$ matrix obtained computing f elementwise.

$\equiv$ A shorthand notation indicating that the quantity on its left-hand side is defined following the expression on its right-hand side.

$\longleftarrow$ A shorthand notation indicating that the quantity on its left-hand side is updated computing the expression on its right-hand side.

$\propto$ A shorthand notation indicating that the quantity on its left-hand side is equal to that on its right-hand side, up to multiplicative constants that can be discarded.

$\approx$ A shorthand notation indicating that the quantity on its left-hand side is approximately equal to that on its right-hand side.

$\ll$ A shorthand notation indicating that the quantity on its left-hand side is much less than that on its right-hand side.

$\mathcal{O}(\cdot)$ For f and g real valued functions, both defined on some unbounded set of real positive numbers and g strictly positive, $f(x) = \mathcal{O}(g(x))$ as $x \to \infty$ if and only if there exist $M \in \mathbb{R}_+$ and $x_0 \in \mathbb{R}$ such that $|f(x)| \leq Mg(x)$ for all $x \geq x_0$.

$\nabla \cdot$ For a scalar-valued function f with arguments $\boldsymbol{x} \in \mathbb{R}^d$, $\nabla f(\boldsymbol{x})$ denotes the associated $d \times 1$ derivative vector.

$\mathsf{H} \cdot$ For a scalar-valued function f with arguments $\boldsymbol{x} \in \mathbb{R}^d$, $\mathsf{H} f(\boldsymbol{x})$ denotes the associated $d \times d$ Hessian matrix.

$\mathfrak{p}(\cdot)$ A generic probability density function.

$\mathfrak{q}(\cdot)$ A generic approximate posterior density function.

$\overset{\text{ind}}{\sim}$ For a sequence generic random variable $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$, $\boldsymbol{x}_i \overset{\text{ind}}{\sim} D_i$ means that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ are independently distributed as $D_1, \ldots, D_d$.

$\overset{\text{iid}}{\sim}$ For a sequence generic random variable $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$, $\boldsymbol{x}_i \overset{\text{iid}}{\sim} D$ means that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ are independently and identically distributed as D.

$E(\cdot)$ For a generic random variable x, $E(x)$ returns its expected value. Usually for a univariate random variable x, $\mu = E(x)$; for a multivariate random vector $\boldsymbol{x}$, $\boldsymbol{\mu} = E(\boldsymbol{x})$; for a matrix-variate random matrix $\boldsymbol{X}$, $\boldsymbol{M} = E(\boldsymbol{X})$.

$\text{Var}(\cdot)$ For a generic univariate random variable x, $\text{Var}(x)$ returns its variance. It is usually denoted with $\sigma^2 = \text{Var}(x)$.

$\text{Cov}(\cdot)$ For a multivariate d-variate random vector $\boldsymbol{x}$, $\text{Cov}(\boldsymbol{x})$ returns its $d \times d$ covariance matrix. It is usually denoted with $\boldsymbol{\Sigma} = \text{Cov}(\boldsymbol{x})$.

$\mathbb{1}(\cdot)$ The indicator function. For a logical condition $\mathcal{P}$, it is such that $\mathbb{1}(\mathcal{P}) = 1$ if $\mathcal{P}$ is true and $\mathbb{1}(\mathcal{P}) = 0$ otherwise.

$\delta(\cdot)$ The *Dirac delta* function. It is defined as the function such that $\int_{-\infty}^{\infty} f(x)\delta(x-c)\mathrm{d}x = f(c)$, for any f and $c \in \mathbb{R}$.

$\phi(\cdot)$ If x is a scalar, $\phi(x)$ represent the probability density function of a univariate standard Gaussian distribution. If $\boldsymbol{x}$ is a vector of dimension $d \times 1$, $\phi(\boldsymbol{x})$ represent the probability density function of a multivariate d-variate standard Gaussian distribution.

$\Phi(\cdot)$ The cumulative distribution function of a standard univariate Gaussian distribution.

$\Gamma(\cdot)$ The *Euler's Gamma* function.

For a scalar x, it is such that: $\Gamma(x) = \int_0^{\infty} u^{x-1}e^u du$.

$\psi(\cdot)$ The first logarithmic derivative of the *Euler's Gamma* function, also known as *digamma* function.

For a scalar $x > 0$, it is such that: $\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x}\log\Gamma(x)$.

$\text{logit}(\cdot)$ The *logit* function.

For a scalar $x \in (0,1)$, it is such that: $\text{logit}(x) = \log\left(\frac{x}{1-x}\right)$.

$\text{expit}(\cdot)$ The *expit* function, usually indicated with $\text{logit}^{-1}(\cdot)$ as it represents the inverse logit function.

For a scalar x, it is such that: $\text{expit}(x) = \frac{1}{1+\exp(-x)}$.

Subscripts are added to give additional informations on how the symbol is assumed to be working, when it is not implicit by the context. Chapter-specific additional notation is described along with the PhD thesis. Notation overlapping between different chapters has to be intended chapter-specific.

Introduction

The world is one big data problem.
There's a bit of arrogance in that,
and a bit of truth as well.
— A. P. McAfee (2014)

Overview

As statisticians and young researchers, we often get stuck in finding credible and easy-to-explain reasons that allow us to motivate the importance of what we study to a broad audience of listeners. To do so, we usually bring up antiquated real-data examples, complicated concepts, and bizarre quotes such as the overused "*I keep saying that the sexy job in the next ten years will be statisticians*" from Google's chief economist Hal Varian. This confounds the ideas and makes us appear as scientists working with odd formulas and incomprehensible graphs.

We are afraid this PhD thesis is not exempt from such contents. However, we would like to introduce it with the main motivations that prompted us to deepen into the fascinating world of approximate variational methods for Bayesian inference, employing an up-to-date example that concerns all of us.

The ongoing *COVID-19* global pandemic of coronavirus disease has changed our lifestyles and confronted us with the urge of being able to extract useful information from the large amount of data that suddenly arose from hospitals, hospitalization facilities, laboratory experiments, and national surveillance systems. Related issues that suddenly occurred can be summarized into three main topics:

1. each day, a wide amount of new data emerges from disparate data sources and need to be stored efficiently into organized structures;

2. statistical software must be scaled to accommodate for immediately extracting real-time information that is useful for countering the evolution of the pandemic;

3. outputs of the analysis need to be statistically reliable and an as informative as possible summary of the large data available.

It is years since statisticians have been challenged by the need to develop methodological tools for addressing these issues. With extreme confidence, we can say that finding adequate and widely accepted solutions will be one of the controversial research trends in the following decades. Collecting data is no longer the main problem, and massive amounts can be produced and stored at a much cheaper cost. Therefore, the urge to develop effective methods for accurate analysis of the so-called *big data* will be highly predominant in the future. This appeared highly prominent in recent times due to monitoring and contrasting the pandemic spread; however, complementary fields in which *big data problems* have arisen for years are, e.g., genomics, neuroscience, economics, physics, medicine, and engineering.

Achieving the aforementioned goals requires a shift in paradigm and statistical thinking (Efron and Hastie, 2016). The speed at which raw and considerable amounts of data are transformed into information is one of the benchmarks on which statistical software are assessed, and it is just as important as the quality of the produced outputs. Therefore, effective statistical procedures must handle large sample sizes by balancing statistical accuracy, easiness of implementation, and computational efficiency.

This PhD thesis presents some developments on *variational approximation methods*. They represent a class of techniques for performing approximate deterministic inference over complex statistical models, which is extremely useful in large data contexts. We will take a Bayesian perspective: the model parameters are equipped with pre-specified distributions encapsulating possibly prior beliefs, and the model likelihood is updated accordingly to the well-known *Bayes theorem* into the so-called posterior distribution of the model. Variational approximation methods determine approximations of the posterior distribution, which are faster to obtain than standard Markov chain Monte Carlo (MCMC) methods, together with assessing a satisfactory degree of accuracy in the approximation. Such approximations are then employed in place of the posterior distribution for taking inferential conclusions.

For decades, the dominant paradigm for performing approximate Bayesian inference when intractable posterior distributions arise has been MCMC (Hastings, 1970; Gelfand and Smith, 1990; Robert and Casella, 2004). It focuses on constructing

an ergodic Markov chain on the full parameter space whose stationary distribution is the posterior distribution and approximates it with samples collected from that chain. Programming languages and ad-hoc software such as WinBUGS (Lunn *et al.*, 2000), JAGS (Plummer *et al.*, 2003) or the more recent Stan (Carpenter *et al.*, 2017) and NIMBLE (de Valpine *et al.*, 2017) helped the diffusion of such approach also for non-practitioners. They circumvent the urge of investigating and manually implementing numerically stable and convergence-guaranteed efficient sampling schemes, only requiring the user to specify the statistical model of interest. Although providing accurate approximations of the posterior distribution, MCMC methods are well known to be computationally demanding and do not scale efficiently with large data scenarios. These main issues gave rise to the diffusion of variational approximation methods for Bayesian inference.

The name has its roots in the mathematical discipline of *calculus of variations* (Gelfand and Fomin, 1963), which focuses on optimizing a functional over a class of functions on which that functional depends. Approximate solutions then arise when the class of functions is restricted in some way to enhance tractability. Due to their remarkable attractiveness, variational methods rapidly became popular in Bayesian statistics literature following two parallel yet separate tracks. Peterson and Anderson (1987) and Opper and Winther (1997) first described the variational procedure for a neural network, and together with the theory from the physical field of statistical mechanics (Parisi, 1988) led to a flurry of approximating procedures for a wide class of other models. At the same time, an algorithm for a similar neural network model was proposed by Hinton and van Camp (1993). Variational approximation methods then became popular in the machine learning field (e.g. Jordan *et al.*, 1999; Jordan, 2004; Titterington, 2004; Bishop, 2006; Wainwright and Jordan, 2008) for addressing elaborate problems such as natural language processing, speech recognition, computational biology, computational neuroscience, computer vision, and robotics. Noteworthy connections to the expectation-maximization (EM) algorithm of Dempster *et al.* (1977) have also been made by Neal and Hinton (1998).

The first introduction to variational approximations using terms from a pure statistical vocabulary is due to Ormerod and Wand (2010), in which they also illustrated with explicit examples how these methods work. Blei *et al.* (2017) is a more recent reference addressing the same urge of disseminating variational approximation inference to statisticians. Zhang *et al.* (2019), instead, is a recent review about the state of the art of variational inference research: references therein give a comprehensive overview of statistical advances in this field. Nonetheless, all these works

mostly focus on variational Bayes approximations, which is the most widespread type of variational approximation method, but not the only one. More exhaustive references concerning variational approximations in general are, e.g., Chapter 10 of Bishop (2006) and Chapters 21 and 22 of Murphy (2012).

Theoretical statistical properties of variational approximation methods from both Bayesian and frequentist perspective have been abundantly studied, see e.g. Hall *et al.* (2002), Wang and Titterington (2006), Hall *et al.* (2011a), Hall *et al.* (2011b), Celisse *et al.* (2012) and You *et al.* (2014) among many others. Two more recent and promising works are Wang and Blei (2019) and Zhang and Gao (2020), in which they established frequentist consistency and asymptotic normality for variational Bayes methods, and studied convergence rates of variational posteriors for nonparametric and high-dimensional inference.

Following Ormerod and Wand (2010), we classify variational approximation methods into two main approaches: the so-called *tangent transform* approach and the alternative *density transform* approach. The former is based on *tangent-type* representations of convex functions and is underpinned by *convex duality* theory (Rockafellar, 1970). The latter involves methods based on approximating the posterior density function with another density function for which the inference is more tractable. This PhD thesis only focuses on the latter type.

Density transform variational methods solve an optimization problem concerned with finding the (optimal) approximate density function which should be the most similar to the posterior density function among a pre-specified family of possible competitors. Variational methods stand out one from each other accordingly to the choices of the following components:

- a divergence $\mathcal{D}$, an information-theoretical measure of proximity between two densities which establishes how *distant* a generic approximate density function is from the model posterior density function. Although most variational approximations are based on the popular Kullback-Leibler divergence (Kullback and Leibler, 1951), many alternative divergences are currently studied;

- a family $\mathcal{Q}$ of approximate density functions over which the optimal approximate density function has to be found. Its structure manages the complexity of the optimization; it is usually chosen to be flexible enough to capture densities close to the posterior but sufficiently simple to conduct efficient optimization. Choices range from non-parametric to parametric specifications, depending on the problem to be addressed;

- the method for finding the optimal solution. Although the optimization problem to be solved is deterministic, it can be addressed both with coordinate ascent algorithm iterating fixed updates until convergence or with stochastic-gradient based approaches, see e.g. Hoffman *et al.* (2013).

Combinations of specific choices for these elements lead to different variational methods. For example, minimization of the Kullback-Leibler divergence leads to the widest adopted class of Variational Bayes (VB) approximations. Among these, if $\mathcal{Q}$ is the family of multivariate Gaussian distributions, then Gaussian Variational Approximations (GVA) method emerges. Conversely, suppose $\mathcal{Q}$ is defined as the set of probability density functions which factorize accordingly to a pre-specified partition of the parameter set: in that case, Mean Field Variational Bayes (MFVB) approximations arise. Their name is inspired by the so-called *mean field* approximations developed in statistical physics (Parisi, 1988). If a mean-field factorization is adopted for $\mathcal{Q}$ having a fixed parametric specification for each of the probability density functions over which its generic element factorizes, and the Kullback-Leibler divergence is used in its reversed order, then Expectation Propagation (EP) approximations are obtained. The literature on variational approximations is tremendously vast and enlarges day-by-day: alternative and more recent approaches are mentioned along with this PhD thesis.

We emphasize that MCMC and variational methods are different approaches to address the same problem. MCMC runs a Markov chain and approximates the posterior distribution with those samples, while variational algorithms solve an optimization problem and approximate the posterior distribution with the optimization result. Nevertheless, if the MCMC sampling procedure is properly designed and the convergence towards the stationary distribution is correctly assessed, it produces samples from the correct posterior distribution of the model. Conversely, variational approximation methods only determine approximations of the true posterior distribution. For this reason, we adopt standard variational literature custom and refer along with this PhD thesis to MCMC methods as those producing the *exact* posterior distribution.

Main Contributions of the Thesis

This PhD thesis presents some developments for different variational approximation methods, mixing the aforementioned three components to develop and investigate novel variational approximation procedures for different statistical models of interest. It is organized into four main chapters.

Chapter 1 presents the mathematical details of variational approximation methods. In particular, it is focused on MFVB, EP, and GVA approximations, which are the most widespread approaches and those upon which the subsequent chapters are built. A detailed presentation would exceed the scope of this work and can be found in references associated with each method. For this reason, the focus is on explaining how these methods can be implemented for a generic statistical model and how the underlying optimization problem is solved.

Chapter 2 deepens into the idea of employing divergences alternative to the Kullback-Leibler. In particular, it focuses on Power-Expectation Propagation (Power-EP), one fruitful generalization of EP arising when the family of α-divergences of Amari (1985) is employed in place of the Kullback-Leibler divergence. The primary motivating idea is that of exploiting the methodology of Minka (2005) for solving the associated optimization problem over a generic factor graph representation of the statistical model of interest: this aids modularization and a generic operational algorithmic structure for approximate inference over arbitrarily complex models. Kim and Wand (2016) reformulated the results by Minka (2005) into a more coherent Bayesian statistical framework, although limiting themselves to developing EP approximations obtained minimizing the α-divergence indexed by $\alpha = 1$. Doing this, they did not benefit from the infinitely uncountable set of optimal approximating density functions that can be found selecting alternative α values. We address this specific issue, developing explicit algorithms for solving Power-EP approximations on some common statistical models, generalizing their EP approximations and encapsulating them as particular cases among a plethora of alternative approximations. In particular, we focus on the subset of α-divergences indexed by $\alpha \in (0, 1]$ since they conceptually represent all those lying in between the Kullback-Leibler divergence used for MFVB approximations and the reverse Kullback-Leibler divergence used for EP approximations. This allows us to study the benefits and limitations of this new class of approximations and exploit new technicalities required to minimize this more general family of divergences. By doing so, we highlight some technical improvements of the algorithms proposed by Kim and Wand (2018) for

performing EP approximations over linear and generalized linear models.

Chapter 3 is devoted to deepening into fixed effects selection procedures on Gaussian-response multilevel models based on streamlined MFVB approximations. MFVB approximations for linear mixed models have already been extensively studied in Section 2.2.3 of Ormerod and Wand (2010) and Section 2.2 of Luts *et al.* (2014), admitting straightforward generalizations to support MFVB fitting of longitudinal and multilevel data. This and many other well-known mixed model specifications account for random effects structures with two-, three- and possible higher-level of nesting that are well known to introduce sparse design matrix structures in the model definition and associated estimation procedures, both in frequentist and Bayesian settings. Lee and Wand (2016) first noticed that MFVB fitting for this class of models employing naïve matrix inversions is problematic for large model dimensions, due to the iterative variational parameter updates involving extremely sparse *ridge regression*-type matrices. They addressed this issue proposing a streamlined variational enhancement for a particular version of two-level mixed models ensuring significantly lower computational efforts, memory usage and required time to assess the same global optima. The streamlined variational methodology has then been extensively studied in Nolan *et al.* (2020) making explicit usage of mathematical results by Nolan and Wand (2020) for efficient solutions of sparse matrix problems arising when dealing with general two-level and three-level random effects specifications. Our work expands and generalizes their methodology and associated variational algorithms to account for a richer family of prior distributions over the fixed effects parameter vector, in contrast to their Gaussian conjugate specification. We focus our study on variational approximations for global-local shrinkage prior specifications, which are well known to handle suitable Bayesian variable selection procedures correctly. By doing so, we extend the streamlined variational methodology to account for their specification over a subset of fixed effects which possibly includes irrelevant covariates. We show with simulated and real data examples that our developed algorithms still provide the optimal results guaranteed by streamlined updating, both in terms of approximation accuracy, memory storage saving, and computational time. We also assess automated tuning-free procedures for fixed effects selection only relying on the obtained variational approximations.

Chapter 4 proposes a novel approach based on GVA approximations for fitting nonstationary Gaussian process regression models. They represent a widely-adopted nonparametric Bayesian method for solving regression problems, which extends classical Gaussian process regression (Rasmussen and Williams, 2006) into

a more general framework that accounts for nonstationary data patterns. We consider a nonstationary Gaussian process regression model following the same approach described by Dunlop *et al.* (2018) for characterizing a deep Gaussian process, although limiting ourselves considering a single latent layer, which in turn takes its foundation on the popular nonstationary kernel construction of Paciorek (2003) and Paciorek and Schervish (2006). A fully-Bayesian inferential approach is adopted, and we approximate the true posterior distribution employing a particular version of GVA having a parsimonious factor structure of the associated covariance matrix, as proposed by Ong *et al.* (2018). We show by simulated data experiments in the unidimensional scenario that satisfactory results can be obtained in terms of model fitting and prediction, in comparison to standard stationary approaches. This opens up new interesting opportunities to be developed for spatial and hyperspatial data settings, although not explicitly experimented in our work. We also outline some possible strategies for making the approximation algorithm efficient in terms of computational timings when the data sample size increases.

Four appendices complete this PhD thesis. Appendix A contains useful information accounting for vector differential calculus, probability distributions (with a specific focus on those belonging to the exponential family), and hardware setup (including programming languages and related libraries) used throughout this PhD thesis. Appendices B, C and D reflect the structure of this PhD thesis and contains supplementary derivations and definitions for Chapters 2, 3 and 4, respectively.

Chapter 1

Variational Approximations for Bayesian Inference

1.1 Bayesian Inference in a Nutshell

In this chapter we summarize the essential key points of Bayesian statistics and variational approximations. In so doing, we hope to make the reading of this PhD thesis accessible to those being unfamiliar with this inferential paradigm, and we also fix the general concepts and notations used hereafter.

Bayesian statistics is a theoretical paradigm for statistical inference based on the Bayesian interpretation of probability expressed as *degree of belief* for a specific event. Unlike the frequentist inference paradigm, here the model parameters are assumed to be stochastic and therefore possess proper distributions. Bayesian statistics updates possible a-priori subjective beliefs about parameters, i.e., prior probability claims, with the observed data information that is assumed to be generated from a statistical model. The updated distribution is then used for inferential reasoning about the model parameters from a probabilistic perspective (Robert, 1994).

Let $\boldsymbol{y}$ be a generic *observed data* vector which is assumed to be generated from a statistical model having *likelihood* $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta})$, and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ be a generic *parameter vector* taking values in the *parameter space* $\boldsymbol{\Theta}$. Without loss of generality, we only treat hereafter the generic case in which both $\boldsymbol{y}$ and $\boldsymbol{\theta}$ are assumed to be continuous. Let $\mathfrak{p}(\boldsymbol{\theta})$ be the *prior* probability density function, summarizing all the a-priori beliefs about model parameters. The main character in Bayesian statistics is the Bayes theorem, which allows to update the prior density function to account for the observed data

likelihood into the *posterior* density function:

$$\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta})\mathfrak{p}(\boldsymbol{\theta})}{\mathfrak{p}(\boldsymbol{y})} = \frac{\mathfrak{p}(\boldsymbol{y},\boldsymbol{\theta})}{\mathfrak{p}(\boldsymbol{y})},$$

where $\mathfrak{p}(\boldsymbol{y},\boldsymbol{\theta}) = \mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta})\mathfrak{p}(\boldsymbol{\theta})$ is the model *joint density* function and $\mathfrak{p}(\boldsymbol{y})$ is the *marginal likelihood* defined as

$$\mathfrak{p}(\boldsymbol{y}) = \int_{\Theta} \mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta})\mathfrak{p}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}.$$

In the following, we interchangeably use terms such as posterior probability density function, posterior distribution, posterior and, equivalently, prior probability density function, prior distribution, prior: the most appropriate term for each occurrence simply derives implicitly from the context in which it is used.

The exact determination of $\mathfrak{p}(\boldsymbol{y})$ requires the marginalization over the parameter space $\boldsymbol{\Theta}$, whose dimensions make this task impractical in almost all situations. As a consequence, a closed analytical expression for $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ is not achievable for most statistical models, except when $\dim(\boldsymbol{\Theta})$ is very low and prior distributions being *conjugate* to the model likelihood are specified.

This main drawback overshadowed the usefulness of Bayesian statistics until recent technological advances made powerful computational resources available for performing onerous simulations in a limited time amount. In particular, MCMC methods provided a revolutionary way to sample from the posterior distributions regardless of the model complexity and parameter dimensions. The essence of MCMC is that of collecting samples $\{\tilde{\boldsymbol{\theta}}^{(t)} : 1 \leq t \leq K\}$ generated from the model posterior distribution, although it is not expressible in a compact and concise form, and using them to (approximately) compute inferential quantities of interest via Monte Carlo integration techniques. For example, the posterior mean value is computed as:

$$E_{\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})}(\boldsymbol{\theta}) = \int_{\Theta} \boldsymbol{\theta}\,\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{\theta} \approx \frac{1}{K}\sum_{t=1}^{K} \tilde{\boldsymbol{\theta}}^{(t)}.$$

1.1.1 The Gibbs Sampler

Literature on MCMC methods is very vast and is still the subject of prolific research. Among many, the *Gibbs sampling* strategy (Casella and George, 1992) has been extensively studied and used for a wide variety of statistical models and still provides useful MCMC sampling strategies when hierarchical Bayesian models with

conjugate prior specifications are of interest. Its attractiveness is due to the possibility of sampling draws from the model posterior distribution simply simulating sub-blocks of the parameters. Its use is particularly appropriate for situations in which there is a convenient partition of the model parameter vector $\boldsymbol{\theta}$ (eventually augmented with *latent* parameters) into M sub-vectors $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$, so that all the associated full conditional posterior densities $\mathfrak{p}(\boldsymbol{\theta}_i|\text{rest})$ belong to notable probability distributions for which efficient sampling routines exist. Hereafter, we denote with "rest" the set containing $\boldsymbol{y}$ and all the $M-1$ partitioned sub-vectors $\boldsymbol{\theta}_j$ for which $j \neq i$. Hence, the Gibbs sampler initializes all the $\boldsymbol{\theta}_i$s to some initial random guess $\tilde{\boldsymbol{\theta}}_i^{(0)}$ given as input, for all $1 \leq i \leq M$, and then performs $B + K \times thin$ subsequent iterations of the following generic t-th step:

1. $\tilde{\boldsymbol{\theta}}^{(t)} \longleftarrow \tilde{\boldsymbol{\theta}}^{(t-1)}$;

2. For $1 \leq i \leq M$:

$$\text{Draw } \tilde{\boldsymbol{\theta}}_i^{(t)} \text{ from } \mathfrak{p}\left(\boldsymbol{\theta}_i \,\middle|\, \boldsymbol{y}, \boldsymbol{\theta}_1^{(t)}, \ldots, \boldsymbol{\theta}_{i-1}^{(t)}, \boldsymbol{\theta}_{i+1}^{(t)}, \ldots, \boldsymbol{\theta}_M^{(t)}\right).$$

As with standard MCMC sampling procedures, the first B collected samples are discarded as *burnin* iterations required for letting the underlying Markov chain reach its stationary distribution. The remaining $K \times thin$ samples are stored one every *thin* to perform a *thinning* procedure of size *thin* reducing possible autocorrelation between subsequent samples. K samples finally remain as output and can be interpreted as being effectively generated from the model posterior distribution.

Although MCMC methods are well-known to asymptotically approximate the true posterior distribution with samples collected from it, they suffer many pitfalls in practical implementations. In fact, they can usually be computationally intensive for moderately complex Bayesian models, with poor mixing and slow convergence being the most common issues compromising the convergence to the stationary distribution, and therefore the validity of inferential conclusions. Moreover, they are well-known to scale difficultly with large sample size datasets. Gibbs sampling techniques are a reasonable solution to make MCMC algorithms faster. However, their implementation is not always possible for all statistical models or perhaps it requires additional computationally-onerous steps such as sampling from complicated distributions or introducing auxiliary variables in the model specification.

1.1.2 Variational Approximations

This PhD thesis focuses on variational approximations, which represent deterministic methods for approximating the posterior distribution that usually exhibits better performances than standard MCMC methods both in terms of computational runtime and scalability. Their main difference with MCMC methods is that they do not rely on direct sampling from the true posterior distribution but instead approximate it with a manageable probability distribution. Following the classification of variational approximation methods mentioned in the introduction of this PhD thesis, we concentrate hereafter on the so-called density transform approach.

Let $\mathcal{D}\{\cdot \,\|\, \cdot\} : \mathbb{S} \times \mathbb{S} \longrightarrow \mathbb{R}^+$ be a generic divergence measure, defined as a function that measures how *distant* two generic probability distributions $\mathfrak{p}, \mathfrak{q} \in \mathbb{S}$ belonging to a generic family of probability density functions $\mathbb{S}$ are from each other over a specific statistical manifold, $\mathcal{D}\{\mathfrak{q} \,\|\, \mathfrak{p}\}$ to be read "the $\mathcal{D}$-divergence of $\mathfrak{p}$ from $\mathfrak{q}$". It represents a weaker notion of distance which best suits for quantifying the dissimilarity between two probability distributions, since in general $\mathcal{D}\{\mathfrak{q}\|\mathfrak{p}\} \neq \mathcal{D}\{\mathfrak{p}\|\mathfrak{q}\}$ and the triangular inequality does not hold. Moreover, let $\mathcal{Q}$ be a generic set of probability density functions that is assumed to contain a sufficiently adequate approximation of $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$, and denote $\mathfrak{q}(\boldsymbol{\theta}) \in \mathcal{Q}$ its generic element. The family $\mathcal{Q}$ can be selected to be very general, following non-parametric specifications, or restricted to some pre-specified parametric family of distributions.

All the variational approximation methods investigated in this PhD thesis can be expressed in terms of the following optimization problem focused on finding the element of $\mathcal{Q}$ being the most similar to $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$, namely:

$$\mathfrak{q}^*(\boldsymbol{\theta}) = \underset{\mathfrak{q}(\boldsymbol{\theta}) \in \mathcal{Q}}{\arg\min}\, \mathcal{D}\{\mathfrak{q}(\boldsymbol{\theta}) \| \mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\}. \tag{1.1}$$

The resulting $\mathfrak{q}^*(\boldsymbol{\theta})$ is the *optimal approximating density function*, or *optimal* $\mathfrak{q}$*-density*, to $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$. It represents the best approximation to the posterior density function that is achievable among the pre-specified set $\mathcal{Q}$ of candidate approximate density functions.

Both the quality of the approximation and the feasibility of the optimization problem depend on the algebraic tractability of $\mathcal{Q}$. Precise approximations arise when a very wide and general family of approximating density functions is selected, but this leads to possible inefficient and computationally demanding algorithms for individuating the optimal density function. On the other hand, way too simplistic choices for $\mathcal{Q}$ lead to poor and useless approximations. Moreover, different choices

of $\mathcal{D}$ identify possibly different optimal solutions and require *ad-hoc* strategies for making the optimization problem effectively solvable.

Different combinations of $\mathcal{D}$, $\mathcal{Q}$ and strategies for solving the optimization problem lead to a very wide set of variational approximation methods, and many others are currently under development. In the remainder of this chapter, we give insights into the three most common variational approximation methods that constitute the keystones on which the next chapters are developed: Mean Field Variational Bayes, Expectation Propagation, and Gaussian Variational Approximations. They all rely upon $\mathcal{D}$ selected as being the popular *Kullback-Leibler divergence*. Recall for two generic probability density functions $\mathfrak{p}_1(\boldsymbol{x})$ and $\mathfrak{p}_2(\boldsymbol{x})$ on $\mathcal{X}$, it is defined as:

$$\mathcal{KL}\{\mathfrak{p}_1(\boldsymbol{x})\|\mathfrak{p}_2(\boldsymbol{x})\} = \int_{\mathcal{X}} \mathfrak{p}_1(\boldsymbol{x}) \log\left\{\frac{\mathfrak{p}_1(\boldsymbol{x})}{\mathfrak{p}_2(\boldsymbol{x})}\right\} \mathrm{d}\boldsymbol{x}.$$

1.2 Mean Field Variational Bayes

Before describing Mean Field Variational Bayes, we introduce the generic family of *Variational Bayes (VB)* approximations as those solving the following optimization problem:

$$\mathfrak{q}^*(\boldsymbol{\theta}) = \underset{\mathfrak{q}(\boldsymbol{\theta})\in\mathcal{Q}}{\arg\min}\, \mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta})\|\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\}. \tag{1.2}$$

They represent the most common approach for variational approximations, due to the possibility of explicitly determining a lower-bound for the model marginal likelihood $\mathfrak{p}(\boldsymbol{y})$ which makes the associated optimization problem tractable. In fact, it is possible to show (Ormerod and Wand, 2010) that:

$$\mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta})\|\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\} = \log \mathfrak{p}(\boldsymbol{y}) - \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta})), \tag{1.3}$$

where $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta})) \equiv E_{\mathfrak{q}(\boldsymbol{\theta})}\{\log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})\} - E_{\mathfrak{q}(\boldsymbol{\theta})}\{\log \mathfrak{q}(\boldsymbol{\theta})\}$ is the *lower bound* on the marginal likelihood $\mathfrak{p}(\boldsymbol{y})$, also known as the *approximate marginal log-likelihood* or the *Evidence Lower BOund (ELBO)*, see Blei *et al.* (2017). Exact computation of (1.3) is mandatory for solving (1.2) but requires the determination of $\mathfrak{p}(\boldsymbol{y})$, a task which we already defined as problematic in almost all practical situations and is supposed to be circumvented by approximate Bayesian methods.

Nevertheless, basic properties of divergences ensure $\mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta})\|\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\} \geq 0$, with the theoretical limiting case $\mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta})\|\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\} = 0$ to be reached if and only if $\mathfrak{q}(\boldsymbol{\theta}) = \mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ almost everywhere, although in practical situations $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}) \notin \mathcal{Q}$. From (1.3) and noticing $\mathfrak{p}(\boldsymbol{y})$ is unknown but constant for each choice of $\mathfrak{q}(\boldsymbol{\theta})$, minimization

of $\mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta})\|\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\}$ over $\mathcal{Q}$ is then equivalent to the maximization of $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}))$ over $\mathcal{Q}$, and a more tractable form for (1.2) is:

$$\mathfrak{q}^*(\boldsymbol{\theta}) = \underset{\mathfrak{q}(\boldsymbol{\theta})\in\mathcal{Q}}{\arg\max}\ \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta})). \tag{1.4}$$

The term "lower bound" arises from the fact that $\log \mathfrak{p}(\boldsymbol{y}) \geq \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}))$ for all $\mathfrak{q}(\boldsymbol{\theta}) \in \mathcal{Q}$. Problem (1.4) is more tractable than (1.2) because an explicit expression for the objective function now only requires the determination of $E_{\mathfrak{q}(\boldsymbol{\theta})}\{\log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})\}$ and $E_{\mathfrak{q}(\boldsymbol{\theta})}\{\log \mathfrak{q}(\boldsymbol{\theta})\}$. Nevertheless, their computation would make VB methods useless in almost all practical situations because it still requires integrating over the parametric space $\boldsymbol{\Theta}$.

Tractability of VB is achieved imposing a suitable form for $\mathcal{Q}$ known as *mean-field restriction*. Let $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ be a pre-specified convenient partition of $\boldsymbol{\theta}$ into M subvectors: *Mean Field Variational Bayes (MFVB)* approximations solve the optimization problem (1.2) under the following additional non-parametric assumption for $\mathcal{Q}$:

$$\mathcal{Q} = \left\{ \mathfrak{q}(\boldsymbol{\theta}) : \mathfrak{q}(\boldsymbol{\theta}) = \prod_{i=1}^{M} \mathfrak{q}_i(\boldsymbol{\theta}_i) \text{ for some partition } \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\} \text{ of } \boldsymbol{\theta} \right\}. \tag{1.5}$$

Notice a very fine partition specified for $\boldsymbol{\theta}$ imposes further approximate posterior independence between the $\mathfrak{q}_i(\boldsymbol{\theta}_i)$s that may not be present in the target $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$. On the other hand, too conservatory partitions would not afford the tractability of the optimization. A trade-off emerges between the accuracy of the obtained approximations and the tractability of MFVB variational approximations. Depending on the amounts of true posterior dependence, the quality of the approximation can range from excellent to poor. Further discussion on this topic is given in Section 3.2 of Titterington (2004).

An important concept well described in Section 10.2.5 of Bishop (2006) is that concerning *induces factorizations* for (1.5) that arise from an interaction between the partition assumed for $\boldsymbol{\theta}$ and the conditional independence properties of $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$. They can be easily detected using simple graphical tests over directed acyclic graph representations of Bayesian models, such as *d-separation theory* (Pearl, 1988). Assume as an illustrative example that $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \boldsymbol{\theta}_3^T)^T$ is partitioned into $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ and $\{\boldsymbol{\theta}_3\}$, and therefore the generic element of $\mathcal{Q}$ factorizes itself into $\mathfrak{q}(\boldsymbol{\theta}) = \mathfrak{q}_{\{1,2\}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)\mathfrak{q}_3(\boldsymbol{\theta}_3)$. If $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are *conditionally independent* given $\boldsymbol{\theta}_3$ and $\boldsymbol{y}$, then the induced factorization $\mathfrak{q}(\boldsymbol{\theta}) = \mathfrak{q}(\boldsymbol{\theta}_1)\,\mathfrak{q}(\boldsymbol{\theta}_2)\,\mathfrak{q}(\boldsymbol{\theta}_3)$ arise. The key aspect here is that the only assumption being made about the generic approximate posterior density function $\mathfrak{q}(\boldsymbol{\theta})$ is that $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$

is a-posteriori independent from $\boldsymbol{\theta}_3$, or equivalently that possible a-posteriori dependence may not be accounted by the specified approximation. Induced factorizations provide further partitioning in the pre-specified partition for $\boldsymbol{\theta}$, and therefore further independence in the structure of $\mathfrak{q}(\boldsymbol{\theta})$, only emerging *by default* from the model structure and the initial partitioning choice being made.

1.2.1 The Coordinate Ascent MFVB Algorithm

From Result 1 of Ormerod and Wand (2010), it follows that the optimal approximating $\mathfrak{q}$-densities satisfy:

$$\mathfrak{q}_i^*(\boldsymbol{\theta}_i) \propto \exp\left\{E_{\mathfrak{q}^*(\boldsymbol{\theta}/\boldsymbol{\theta}_i)}(\log \mathfrak{p}(\boldsymbol{\theta}_i|\text{rest}))\right\}, \quad 1 \leq i \leq M, \tag{1.6}$$

where $E_{\mathfrak{q}^*(\boldsymbol{\theta}/\boldsymbol{\theta}_i)}(\cdot)$ denotes the expectation with respect to $\mathfrak{q}^*(\boldsymbol{\theta})/\mathfrak{q}_i^*(\boldsymbol{\theta}_i) = \prod_{j\neq i}\mathfrak{q}_j^*(\boldsymbol{\theta}_j)$ and $\mathfrak{p}(\boldsymbol{\theta}_i|\text{rest})$ is the full conditional posterior density function for $\boldsymbol{\theta}_i$.

The maximization of (1.4) is therefore achievable initializing all the $\mathfrak{q}_i(\boldsymbol{\theta}_i)$'s and updating each in turn by replacing its current estimate with a revised expression given by:

$$\mathfrak{q}_i(\boldsymbol{\theta}_i) \longleftarrow \frac{\exp\left\{E_{\mathfrak{q}(\boldsymbol{\theta}/\boldsymbol{\theta}_i)}(\log \mathfrak{p}(\boldsymbol{\theta}_i|\text{rest}))\right\}}{\int \exp\left\{E_{\mathfrak{q}(\boldsymbol{\theta}/\boldsymbol{\theta}_i)}(\log \mathfrak{p}(\boldsymbol{\theta}_i|\text{rest}))\right\}\,\mathrm{d}\boldsymbol{\theta}_i}, \quad 1 \leq i \leq M, \tag{1.7}$$

until the relative increase in $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}))$ from two subsequent iterations is negligible. Once convergence is assumed to be reached, $\mathfrak{q}^*(\boldsymbol{\theta}) = \prod_{i=1}^M \mathfrak{q}_i^*(\boldsymbol{\theta}_i)$ is the solution to the optimization problem (1.4) and represents the optimal MFVB approximation to $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ under product restriction (1.5). Moreover, $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}^*(\boldsymbol{\theta}))$ represents the optimal MFVB approximation to $\log \mathfrak{p}(\boldsymbol{y})$.

Each update (1.7) uniquely minimizes the Kullback-Leibler divergence (or, equivalently, maximizes the lower bound) with respect to $\mathfrak{q}_i(\boldsymbol{\theta}_i)$, forming an iterative coordinate ascent algorithm for climbing the lower-bound, as explained for example in Section 10.1.1 of Bishop (2006) and Section 2.2 of Ormerod and Wand (2010). Convexity properties can be used to show that convergence to at least one local optima is guaranteed (Boyd and Vandenberghe, 2004; Luenberger and Ye, 2008). Interestingly, Beal and Ghahramani (2003) illustrated how the MFVB iterative updating scheme can be interpreted as a generalization of the *Expectation-Maximization* algorithm of Dempster *et al.* (1977). Such procedure was initially designed for finding maximum

likelihood estimates in models with latent variables, which alternates between computing the first addendum of the lower bound $E\{\log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})\}$ according to $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ instead of $\mathfrak{q}(\boldsymbol{\theta})$ (the E-step), assuming the expectation under $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ is computable, and optimizing it with respect to the model parameters (the M-step).

1.2.2 Practicalities

Suppose all the parameters in the model are conditionally conjugate to each other, as it sometimes happens in practical implementations. In that case, the optimal $\mathfrak{q}$-density functions in (1.6) are available in closed parametric forms. Therefore, their iterative updates reduce into iteratively updating its associated parameters using the updated parameters of the remaining $M-1$ approximating densities. A more straightforward implementation arises when all the full conditional distributions $\boldsymbol{\theta}_i|\text{rest}$ belong to the exponential family (Wainwright and Jordan, 2008): in this case the updates are even easier to express in terms of sums of natural parameter vectors, see Section 4.1 of Blei *et al.* (2017). The work by Wand (2017) made explicit usage of this property, deriving fast approximation algorithms for arbitrarily large models with an approach founded upon a message passing formulation of MFVB that utilizes factor graph representations of statistical models, known in the literature as *variational message passing* (Winn and Bishop, 2005). In all the other situations, numerical quadrature techniques are required to approximate the denominator of (1.7). This task is computationally more demanding and affects the benefits of variational approximation procedures.

Practical examples of MFVB approximations for some common statistical models are listed in Section 2.2 of Ormerod and Wand (2010), to which we refer. Chapter 3 of the present PhD thesis uses streamlined MFVB versions for obtaining approximations over linear mixed models and provides practical usage for the formulas presented in the previous pages.

1.3 Expectation Propagation

The idea behind *Expectation Propagation (EP)* was first revealed by Thomas Minka through his PhD thesis (Minka, 2001b) and the resulting seminal paper (Minka, 2001a). It unifies two existing approximation techniques: *assumed-density filtering*

(Maybeck, 1979), a one-pass sequential method for computing approximate posterior distributions, and *loopy belief propagation* (Murphy *et al.*, 1999), an extension of *belief propagation* for Bayesian networks (Frey and MacKay, 1998).

Following Minka (2005), it is possible to express EP approximations as those resulting from the following optimization problem:

$$\mathfrak{q}^*(\boldsymbol{\theta}) = \underset{\mathfrak{q}(\boldsymbol{\theta}) \in \mathcal{Q}}{\arg\min}\, \mathcal{KL}\{\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}) \| \mathfrak{q}(\boldsymbol{\theta})\}. \tag{1.8}$$

Notice EP uses the Kullback-Leibler divergence in its reverse order, usually known as *inclusive Kullback-Leibler* divergence if compared to the one used for defining the MFVB optimization problem in (1.2). Solution of (1.8) is way more involved than the coordinate ascent algorithm required for MFVB, and EP is well-known to uniquely provide a heuristic method for the solution of the associated optimization problem. In fact, few theoretical guarantees that it converges to the optimal solution are present. Nevertheless, it has been shown to outperform alternative variational approximation methods in specific contexts, see e.g. Nickisch and Rasmussen (2008), Jylänki *et al.* (2011), Gehre *et al.* (2014) and Jylänki *et al.* (2014). A general roadmap to research on EP and associated variants can be found in `https://tminka.github.io/papers/ep/roadmap.html`.

A convenient partition of $\boldsymbol{\theta}$ into M different subsets $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ is also employed here, such that $\mathcal{Q}$ assumes the same identical form of (1.5). In addition, EP requires a coherent parametric specification for each $\mathfrak{q}_i(\boldsymbol{\theta}_i)$ to be a member of the exponential family, namely:

$$\mathfrak{q}_i(\boldsymbol{\theta}_i; \boldsymbol{\eta}_{\mathfrak{q}_i(\boldsymbol{\theta}_i)}) = \exp\left\{ \boldsymbol{T}_i(\boldsymbol{\theta}_i)^T \boldsymbol{\eta}_{\mathfrak{q}_i(\boldsymbol{\theta}_i)} - A_i\big(\boldsymbol{\eta}_{\mathfrak{q}_i(\boldsymbol{\theta}_i)}\big) \right\} h_i(\boldsymbol{\theta}_i), \quad \boldsymbol{\eta}_{\mathfrak{q}_i(\boldsymbol{\theta}_i)} \in \mathbb{H}_i,$$

for each $1 \leq i \leq M$. A subscript i indicates that the sufficient statistic vector $\boldsymbol{T}_i$, the log-partition function A_i and the base measure h_i are referred to the exponential family $\mathcal{Q}_i$ to which $\mathfrak{q}_i(\boldsymbol{\theta}_i)$ is restricted to belong. Moreover, $\boldsymbol{\eta}_{\mathfrak{q}_i(\boldsymbol{\theta}_i)}$ denotes the natural parameter vector uniquely identifying $\mathfrak{q}_i(\boldsymbol{\theta}_i)$ among $\mathcal{Q}_i$.

Two main formulations by which EP is defined for approximate Bayesian inference arise in literature. The more prominent and common approach dates back to the seminal paper by Thomas Minka. It assumes $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ comprises a product over so-called *sites*, often corresponding to data points, and so-called *cavity* and *tilted* distributions are obtained by iteratively performing moment-matching procedures. Details are well explained in, e.g., Section 10.7 of Bishop (2006), Section 13.8 of Gelman *et al.* (2014) and Section 2.1 of Vehtari *et al.* (2020). Hereafter, we focus on the

alternative approach motivated by Minka (2005) and Minka and Winn (2008) that finds EP approximations with a *message passing* algorithm on a factor graph representation for the statistical model being considered. This second approach is more general than the original one because different exponential family distributions can be selected for the $\mathfrak{q}_i(\boldsymbol{\theta}_i)$'s. Moreover, it permits us to better understand and digest the work and notation presented in Chapter 2.

1.3.1 Factor Graphs

Factor graphs (Kschischang *et al.*, 2001; Frey, 2002) are a relatively new graphical concept in the statistical field. They graphically represent the dependencies occurring in a precise statistical model, together with the product density form specified for the generic $\mathfrak{q}(\boldsymbol{\theta})$. Moreover, they permit to compartmentalize the algebra and coding implementations required for estimating the optimal approximate posterior density function via a *message-passing* algorithm that can be built in terms of *fragments*. Wand (2017) developed factor graph fragmentization to streamline MFVB approximations over semiparametric regression models, while Chen and Wand (2020) mimicked the same idea for EP approximations.

We describe factor graphs through a self-explanatory naïve example that is useful for fully understanding the content of Chapter 2. Consider a Bayesian model with parameter $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \boldsymbol{\theta}_3^T)^T$ and assume $\boldsymbol{\theta}$ is partitioned into $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$; then, suppose its joint density function $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ can be factorized into the product $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_3)\, \mathfrak{p}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)\, \mathfrak{p}(\boldsymbol{\theta}_2)\, \mathfrak{p}(\boldsymbol{\theta}_3)$. Such model can be visually represented with the factor graph displayed in Figure 1.1, having four different *factors* (represented with full squared boxes) being those onto which $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ factorizes and three different *stochastic nodes* (represented with empty circles) representing those onto which $\boldsymbol{\theta}$ have been partitioned. Factor graphs connect all the stochastic nodes belonging to a specific factor (usually called the *neighborhood* of that factor) with a straight line, allowing to immediately visualize the dependencies among parameters induced by the Bayesian specification of a model. For example, the model likelihood factor $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_3)$ depends upon $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3$: therefore both nodes share a straight line with it, and compose its neighborhood.

Multiple factor graph representations are possible for the same Bayesian model. They depend upon the choice for the partition of $\boldsymbol{\theta}$ and the different factors onto which $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ can be factorized. For more complex statistical models, convenient choices are required to implement feasible message-passing algorithms.

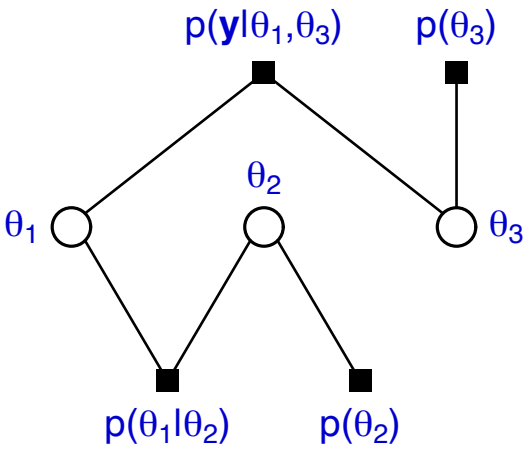

FIGURE 1.1: *Factor graph representation of the naïve Bayesian model described in Section 1.3.1, with* $\boldsymbol{\theta}$ *partitioned into* $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ *and* $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ *factorized into the product* $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_3)\, \mathfrak{p}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)\, \mathfrak{p}(\boldsymbol{\theta}_2)\, \mathfrak{p}(\boldsymbol{\theta}_3)$.

1.3.2 The EP Message-Passing Algorithm

We now present the mathematical details of the general prescription proposed by Minka (2005) for solving (1.8) over a generic factor graph representation for a Bayesian model. To do so, we borrow concepts and notation from the statistically-oriented explanation given in Section 3 of Kim and Wand (2016).

Let then consider the pre-specified partition of $\boldsymbol{\theta}$ into M different sub-vectors, each representing a stochastic node in the model associated factor graph representation, and denote $\boldsymbol{\theta}_S$ the set composed by the $\boldsymbol{\theta}_i$'s having index $i \in S$, S belonging to the power set of $\{1, \ldots, M\}$. Given such a partition, it is possible to factorize the model joint density function into N pieces f_j representing factors in the model associated factor graph representation, namely:

$$\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta}) = \prod_{j=1}^{N} f_j(\boldsymbol{\theta}_{\text{neigh}(j)}),$$

where $\text{neigh}(j)$ is the set of stochastic node indexes belonging to the neighborhood of factor f_j, for each $1 \leq j \leq N$. For example, when considering hierarchical Bayesian model specifications in which $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ admits a *directed acyclic graphical model* representation with *hidden nodes* $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M$, *evidence node* $\boldsymbol{y}$ and arrows indicating conditional dependence relationships among the model random variables, then

$$\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta}) = \mathfrak{p}\left(\boldsymbol{y}|\text{parents}\{\boldsymbol{y}\}\right) \prod_{i=1}^{M} \mathfrak{p}\left(\boldsymbol{\theta}_i|\text{parents}\{\boldsymbol{\theta}_i\}\right),$$

with $N = M + 1$ different factors f_j (M corresponding to the density functions of $\boldsymbol{\theta}_i$

conditional on its parents, $1 \leq i \leq M$, and 1 corresponding to the likelihood factor). Here parents$\{\cdot\}$ denotes the set of parental nodes of a specific node in a graphical model, i.e., the set of hidden nodes pointing an arrow towards it. With direct reference to the factor graph displayed in Figure 1.1, it consists of $M = 3$ stochastic nodes $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3$ from which the joint density function $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ can be factorized into $N = 4$ factors: $\mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_3)$, $\mathfrak{p}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)$, $\mathfrak{p}(\boldsymbol{\theta}_2)$ and $\mathfrak{p}(\boldsymbol{\theta}_3)$.

The EP *message passing* algorithm runs over the resulting factor graph representation of the model, iteratively computing and updating for each non-trivial combination of $1 \leq i \leq M$ and $1 \leq j \leq N$ the *EP stochastic node to factor messages*:

$$\mathfrak{m}_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i) \longleftarrow \prod_{j' \neq j \,:\, i \in \text{neigh}(j')} \mathfrak{m}_{f_{j'} \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \tag{1.9}$$

and the *EP factor to stochastic node messages*:

$$\mathfrak{m}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \longleftarrow \frac{\text{proj}_{\mathcal{Q}_i}\left[\begin{array}{l} Z^{-1}\, \mathfrak{m}_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i) \\ \times \int f_j(\boldsymbol{\theta}_{\text{neigh}(j)}) \prod\limits_{i' \in \text{neigh}(j) \setminus \{i\}} \mathfrak{m}_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})\, \mathrm{d}\boldsymbol{\theta}_{\text{neigh}(j) \setminus \{i\}} \end{array}\right]}{\mathfrak{m}_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i)}, \tag{1.10}$$

where Z is the normalizing constant ensuring the density function in $\boldsymbol{\theta}_i$ inside $\text{proj}_{\mathcal{Q}_i}[\cdot]$ integrates to one. Importantly, $\text{proj}_{\mathcal{Q}_i}[\cdot]$ is the operator performing the *Kullback-Leibler projection* of a general probability density function $z(\boldsymbol{\theta}_i)$ onto the pre-specified family of distributions $\mathcal{Q}_i$ for $\mathfrak{q}_i(\boldsymbol{\theta}_i)$, namely

$$\text{proj}_{\mathcal{Q}_i}[z(\boldsymbol{\theta}_i)] = \underset{\mathfrak{q}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \in \mathcal{Q}_i}{\arg\min}\ \mathcal{KL}\{z(\boldsymbol{\theta}_i) \| \mathfrak{q}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)\}. \tag{1.11}$$

Finding a solution to this optimization problem is generally unfeasible, unless $\mathcal{Q}_i$ belongs to the exponential family of distributions, as it is usually the case in EP for all $1 \leq i \leq M$. In this scenario,

$$\mathfrak{q}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) = \mathfrak{q}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i; \boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i}) = \exp\left\{ \boldsymbol{T}_i(\boldsymbol{\theta}_i)^T \boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i} - A_i\left(\boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i}\right) \right\} h_i(\boldsymbol{\theta}_i)$$

and (1.11) is equivalent of solving the following optimization problem:

$$\boldsymbol{\eta}^*_{f_j \to \boldsymbol{\theta}_i} = \underset{\boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i} \in \mathbb{H}_i}{\arg\min}\ \mathcal{KL}\{z(\boldsymbol{\theta}_i) \| \mathfrak{q}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i; \boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i})\}, \tag{1.12}$$

which essentially reduces to finding the optimal natural parameter vector $\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i}$ such that the following sufficient statistic *moment-matching* equality holds:

$$\int \boldsymbol{T}_i(\boldsymbol{\theta}_i) \exp\left\{\boldsymbol{T}_i(\boldsymbol{\theta}_i)^T \boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i} - A_i\left(\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i}\right)\right\} h_i(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_i = \int \boldsymbol{T}_i(\boldsymbol{\theta}_i) z(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_i.$$

Due to the fact that

$$E_{\mathrm{q}_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i;\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i})}\{\boldsymbol{T}(\boldsymbol{\theta}_i)\} = \nabla_{\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}} A_i\left(\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}\right)$$

for a generic distribution belonging to the exponential family, the above equality can be rewritten as

$$\nabla_{\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}} A_i\left(\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i}\right) = \int \boldsymbol{T}_i(\boldsymbol{\theta}_i) z(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_i,$$

where $\nabla_{\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}} A_i\left(\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i}\right)$ is to be intended as "the derivative vector of $A_i(\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i})$ with respect to $\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}$, computed in $\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i}$". The optimal $\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i}$ is then:

$$\boldsymbol{\eta}^*_{f_j\to\boldsymbol{\theta}_i} = \left\{ (\nabla_{\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}} A_i)^{-1} \left(\int \boldsymbol{T}_i(\boldsymbol{\theta}_i) z(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_i \right) \right\}. \tag{1.13}$$

In other words, when $\mathcal{Q}_i$ is restricted to be a family of distributions belonging to the exponential family, $\mathrm{proj}_{\mathcal{Q}_i}[z(\boldsymbol{\theta}_i)]$ identifies the probability density function among $\mathcal{Q}_i$ whose Kullback-Leibler divergence is minimal with respect to $z(\boldsymbol{\theta}_i)$. The identification simply arises determining the optimal natural parameter vector resulting from the inversion of the derivative vector of $A_i(\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i})$ with respect to $\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}$, computed in $E_{z(\boldsymbol{\theta}_i)}\{\boldsymbol{T}_i(\boldsymbol{\theta}_i)\}$.

For all the regular distributions belonging to the exponential family (i.e., such that $\mathbb{H}_i$ is an open set of $\mathbb{R}^{k_i}$ for a certain $k_i \in \mathbb{N}$) in which $\boldsymbol{T}_i(\boldsymbol{\theta}_i)$ is the minimal sufficient statistic vector, results from Section 3 of Wainwright and Jordan (2008) ensures that $\nabla_{\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}} A_i(\cdot)$ is a bijective map and therefore $(\nabla_{\boldsymbol{\eta}_{f_j\to\boldsymbol{\theta}_i}} A_i)^{-1}(\cdot)$ is well-defined. This is the case, for example, for the univariate and multivariate Normal distributions and for the Inverse-χ^2 distribution expressed in their minimal representation. Explicit examples of the Kullback-Leibler projections for some common exponential families were given in, e.g., Section 2.2 of Kim and Wand (2016) and Section 2.2 of Chen and Wand (2020), and are extensively discussed in Chapter 2.

Both types of messages are, in practice, proportional to proper probability density functions. The EP algorithm initializes all the messages involving $\boldsymbol{\theta}_i$ to be a pre-specified density function in $\mathcal{Q}_i$, for all $1 \leq i \leq M$, and iteratively updates them with the expressions given in (1.9)–(1.10). Upon convergence of the messages, the

optimal $\mathfrak{q}$-densities are obtained with:

$$\mathfrak{q}_i^*(\boldsymbol{\theta}_i) \propto \prod_{j:i\in\text{neigh}(j)} \mathfrak{m}_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i), \quad 1 \leq i \leq M. \tag{1.14}$$

Notice the effect of the $\text{proj}_{\mathcal{Q}_i}[\cdot]$ operator in (1.10) is that of ensuring all the messages starting from $\boldsymbol{\theta}_i$ or pointing towards it to belong to a specific exponential family $\mathcal{Q}_i$. This means that the expressions (1.9)–(1.10) boil down into uniquely updating the natural parameter vectors associated with those messages, and multiplications (divisions) between EP messages boil down to sums (differences) of their natural parameter vectors.

A reasonable stopping criterion is that of monitoring the evolution of the lower bound $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}))$, similarly to what is usually done in MFVB, and stopping when its relative change from one iteration to another becomes negligible. Differently from MFVB, this quantity is not necessarily monotone, nor is convergence guaranteed, as pointed out in Section 10.7 of Bishop (2006). In terms of the notation introduced so far for the factor graph representation of a statistical model, its expression is:

$$\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta})) = \sum_{i=1}^{M} \log s_{\boldsymbol{\theta}_i} + \sum_{j=1}^{N} \log s_{f_j} \tag{1.15}$$

where

$$s_{\boldsymbol{\theta}_i} = \int \prod_{j:i\in\text{neigh}(j)} \mathfrak{m}_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_i \tag{1.16}$$

and

$$s_{f_j} = \frac{\displaystyle\int f_j(\boldsymbol{\theta}_{\text{neigh}(j)}) \prod_{i\in\text{neigh}(j)} \mathfrak{m}_{\boldsymbol{\theta}_i\to f_j}(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_{\text{neigh}(j)}}{\displaystyle\int \prod_{i\in\text{neigh}(j)} \mathfrak{m}_{\boldsymbol{\theta}_i\to f_j}(\boldsymbol{\theta}_i)\, \mathfrak{m}_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)\, \mathrm{d}\boldsymbol{\theta}_{\text{neigh}(j)}}. \tag{1.17}$$

Notice this message-passing approach for finding EP approximations follows the same prescription for variational message passing (Winn and Bishop, 2005). As clarified by Appendixes A and B of Minka and Winn (2008), the only difference between EP and MFVB via message passing resides on the expression for the factor to stochastic node message update (1.10), which does not require any Kullback-Leibler projection over a pre-determined exponential family for $\mathfrak{q}(\boldsymbol{\theta}_i)$ and therefore is simply:

$$\mathfrak{m}_{f_j\to\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \longleftarrow \exp\left\{E_{f_j\to\boldsymbol{\theta}_i}\left[\log f_j(\boldsymbol{\theta}_{\text{neigh}(j)})\right]\right\},$$

where $E_{f_j \to \boldsymbol{\theta}_i}[\cdot]$ denotes the expectation with respect to the density function:

$$\frac{\prod_{i' \in \text{neigh}(j) \setminus \{i\}} \mathfrak{m}_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'})\, \mathfrak{m}_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})}{\prod_{i' \in \text{neigh}(j) \setminus \{i\}} \int \mathfrak{m}_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'}) \mathfrak{m}_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})\, \mathrm{d}\boldsymbol{\theta}_{i'}}.$$

1.3.3 Theoretical Developments

EP approximations have been prevalent in research areas including machine learning and computer science, with less attention from the statistical community until recent years. This came with a wide variety of international conference proceeding papers, technical reports, and *arXiv* pre-prints to be flourishing in the last decades accounting for theoretical developments and practical implementations of EP over many fields. Nonetheless, a limited number of papers have been published in popular statistical journals.

Recently, research on EP approximations has been increasing also in the Bayesian statistics field, and it is worth mentioning the title of Vehtari *et al.* (2020)'s paper published in the statistically-oriented Journal of Machine Learning Research, "Expectation propagation as a way of life: a framework for Bayesian inference on partitioned data" to understand how prominent this line of research will be over the next years. Among recent works studying EP approximations from a purely statistical perspective, we mention Kim and Wand (2016) in which an explicit form for the EP algorithm has been derived for the univariate Normal random sample model and their later Kim and Wand (2018) paper in which the same task was tackled for generalized, linear and mixed-effect regression models. Both works allow to better grasp mathematical details of the derivation and computational challenges required for the practical implementation of EP for such models. Barthelmé and Chopin (2014) extended the idea behind EP to approximate Bayesian computation (ABC) and develop an algorithm that is faster by a few orders of magnitude than standard ABC approaches while producing a negligible approximation error. Dehaene and Barthelmé (2015) presented bounds for the EP approximation error, while Dehaene and Barthelmé (2018) introduced a variant of EP, called *averaged-EP*, and compared both methods in the large data limit to prove that EP is asymptotically exact under certain conditions. Another extension of EP that maintains a global posterior approximation but updates it in a local way, called stochastic expectation propagation, was proposed by Li *et al.* (2015b). EP approximations have also been recently used by Hall *et al.* (2020) to approximate integrals arising in frequentist statistical inference for binary response mixed-effect models, showing that fast and accurate

quadrature-free inference can be realized for the probit link case with multivariate random effects and higher levels of nesting.

Regarding its implementation in standard statistical software, we mention the Infer.NET (Minka *et al.*, 2018) computational framework for approximate Bayesian inference in hierarchical Bayesian models, which implements EP approximations in addition to standard MFVB, although for a limited number of statistical models (see Wang and Wand, 2011, for practical examples).

Chapter 2 of the present PhD thesis investigates approximations for some statistical models with Power-EP, a recently-developed improved version of EP accounting for the minimization of a divergence alternative to the Kullback-Leibler. Practical illustrations and insights into the formulas and notation introduced in the previous pages are exhaustively given.

1.4 Gaussian Variational Approximations

Gaussian Variational Approximations (GVA) are expressible as a particular case of VB approximations described in the first paragraph of Section 1.2, for which $\mathcal{Q}$ is restricted to be the family of multivariate Normal distributions of dimension k, k supposed to be the length of the model parameter vector $\boldsymbol{\theta}$. Therefore, they arise as the optimal solution of a modified version of (1.2) expressible as:

$$\mathfrak{q}^*(\boldsymbol{\theta}; \boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}) = \underset{\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}) \in \mathcal{Q}}{\arg\min} \mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}) \| \mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\}, \tag{1.18}$$

where $\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \phi(\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta} - \boldsymbol{\mu}))$ denotes the probability density function of a multivariate Normal distribution with mean parameter $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Notice GVA is a valid variational approximation procedure if and only if $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^k$; otherwise, model parameters contained in $\boldsymbol{\theta}$ which are restricted to belong to some subsets of $\mathbb{R}$ must be adequately reparametrized.

Since the multivariate Normal distribution is uniquely identified by its mean parameter and covariance matrix, (1.18) can be equivalently expressed into finding $\mathfrak{q}^*(\boldsymbol{\theta})$ being the optimal $\mathrm{N}\big(\boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}\big)$ density function such that:

$$\Big(\boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}\Big) = \underset{(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}) \in \mathbb{R}^k \times \mathbb{S}^k_+}{\arg\min} \mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}) \| \mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\}. \tag{1.19}$$

Following its exponential family representation described in Appendix A.2, resolution of (1.19) is also identical to find

$$\boldsymbol{\eta}^*_{\mathfrak{q}(\boldsymbol{\theta})} = \underset{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} \in \mathbb{H}}{\arg\min}\, \mathcal{KL}\{\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}) \| \mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\}, \tag{1.20}$$

where $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}$ indicates the $(k + k(k+1)/2)$-dimensional natural parameter vector of a multivariate Normal distribution of dimension k belonging the the natural parameter vector space $\mathbb{H}$, from which:

$$\boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})} = -\frac{1}{2}\left\{\text{vech}^{-1}\left([\boldsymbol{\eta}^*_{\mathfrak{q}(\boldsymbol{\theta})}]_2\right)\right\}^{-1} \quad \text{and} \quad \boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})} = \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}[\boldsymbol{\eta}^*_{\mathfrak{q}(\boldsymbol{\theta})}]_1$$

follows. Nevertheless, (1.18) still suffers the same problematics of VB due to the necessity of explicitly computing $\mathfrak{p}(\boldsymbol{y})$: with similar motivations, it is possible to rephrase it in terms of the following optimization problem

$$\begin{aligned} \mathfrak{q}^*(\boldsymbol{\theta}; \boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}) &= \underset{(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}) \in \mathbb{R}^k \times \mathbb{S}^k_+}{\arg\max} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})})) \\ &= \underset{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} \in \mathbb{H}}{\arg\max}\, \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})), \end{aligned} \tag{1.21}$$

that is, of individuating $\boldsymbol{\eta}^*_{\mathfrak{q}(\boldsymbol{\theta})}$ (or equivalently, both $\boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}$ and $\boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}$) that maximizes the marginal likelihood lower bound.

1.4.1 Literature Review

Although it has been summarized here taking a Bayesian perspective, GVA has been extensively used for obtaining variational approximations both from the frequentist and the Bayesian side. Frequentist GVA methods have been discussed in Section 4 of Ormerod and Wand (2010), while Hall *et al.* (2011b) and Hall *et al.* (2011a) used them to estimate the parameters of a mixed-effects linear model with Poisson response. They proved root-m asymptotic consistency and normality of the obtained estimates under relatively mild assumptions, together with providing explicit expressions for their rates of convergence. Heuristic arguments were given by Ormerod and Wand (2012) to extend these results for proving the consistency of GVA for generalized linear mixed models.

Early contributions from a Bayesian perspective are due to Hinton and van Camp (1993) and Barber and Bishop (1998), who used GVA-type approximations for neural networks models. Later works include, e.g.: Archambeau *et al.* (2007), in which

GVA were applied to the posterior measure over paths for a general class of stochastic differential equations; Raiko *et al.* (2007), which used GVA over latent variable models; Nickisch and Rasmussen (2008), who derived GVA for binary Gaussian process classification tasks; Honkela *et al.* (2008), who used the Riemannian structure of the multivariate Normal distribution to derive a more efficient algorithm for GVA, and Opper and Archambeau (2009), in which they showed that GVA methods are extremely efficient in terms of big-$\mathcal{O}$ computations if applied to certain classes of probabilistic models. We refer to Challis and Barber (2013) for an extensive review.

An iterative coordinate ascent algorithm is required for solving (1.21), with many different numerical optimization strategies that can be implemented, as well summarized in Section 3 of Rohde and Wand (2016). A fixed-point iteration update scheme was derived in Wand (2014), with generic update expressions:

$$\begin{cases} \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}^{\text{new}} \longleftarrow -\frac{1}{2}\left\{\text{vech}^{-1}\left(\nabla_{\text{vech}(\boldsymbol{\Sigma}_{q(\boldsymbol{\theta})})} \log \underline{p}(\boldsymbol{y}; q(\boldsymbol{\theta}; \boldsymbol{\mu}_{q(\boldsymbol{\theta})}^{\text{old}}, \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}^{\text{old}}))\right)\right\}^{-1} \\ \boldsymbol{\mu}_{q(\boldsymbol{\theta})}^{\text{new}} \longleftarrow \boldsymbol{\mu}_{q(\boldsymbol{\theta})}^{\text{old}} + \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}^{\text{new}}\left\{\nabla_{\boldsymbol{\mu}_{q(\boldsymbol{\theta})}} \log \underline{p}(\boldsymbol{y}; q(\boldsymbol{\theta}; \boldsymbol{\mu}_{q(\boldsymbol{\theta})}^{\text{old}}, \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}^{\text{old}}))\right\}. \end{cases}$$

The superscripts "old" and "new" distinguish between the older and updated values at each iteration, respectively. Alternative update expressions follow from Result 2 of Rohde and Wand (2016) and only requires the first and second order derivatives of the lower-bound $\log \underline{p}(\boldsymbol{y}; q(\boldsymbol{\theta}; \boldsymbol{\mu}_{q(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}))$ to be computed with respect to $\boldsymbol{\mu}_{q(\boldsymbol{\theta})}$, namely updating $\boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}$ with the following alternative expression:

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}^{\text{new}} \longleftarrow -\left\{\mathsf{H}_{\boldsymbol{\mu}_{q(\boldsymbol{\theta})}} \log \underline{p}(\boldsymbol{y}; q(\boldsymbol{\theta}; \boldsymbol{\mu}_{q(\boldsymbol{\theta})}^{\text{old}}, \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}^{\text{old}}))\right\}^{-1}.$$

Both expressions require the explicit derivation of $\log \underline{p}(\boldsymbol{y}; q(\boldsymbol{\theta}; \boldsymbol{\mu}_{q(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}))$, and when this is not available in closed form stochastic gradient ascent methods are usually employed. We focus on the details of this latter approach, being the one that is used in Chapter 4.

1.4.2 Stochastic Gradient Ascent-Based GVA

Stochastic gradient ascent (Robbins and Monro, 1951) is an iterative method for optimizing an objective function with suitable smoothness properties that have been largely used in variational approximation, see e.g. Paisley *et al.* (2012), Nott *et al.* (2012), Hoffman *et al.* (2013), Salimans and Knowles (2013), Ranganath *et al.* (2014), Titsias and Lázaro-Gredilla (2014) and Kucukelbir *et al.* (2017) among others. In

simple words, stochastic gradient ascent for solving (1.21) starts from an initial value $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^{(0)}$ for $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}$ and proceeds recursively performing the following update:

$$\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^{(t+1)} \longleftarrow \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^{(t)} + \boldsymbol{\rho}_t \circ \left\{ \widehat{\nabla}_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^{(t)})) \right\} \tag{1.22}$$

for $t = 1, 2, \ldots$ until a stopping condition is satisfied. Here $\widehat{\nabla}_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^{(t)}))$ is an unbiased estimate for the gradient vector of $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))$ with respect to $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}$, evaluated in $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^{(t)}$, and $\{\boldsymbol{\rho}_t\}_{t \geq 0}$ is a sequence of vector-values learning rates typically chosen so that its elements satisfy the Robbins-Monro conditions (Robbins and Monro, 1951). A large literature on different adaptive choices of the learning rates exists, see `https://ruder.io/optimizing-gradient-descent/` for a detailed review. Convergence to a local optimum under minor regularity conditions is usually ensured (Bottou, 2010).

Different approaches for the construction of unbiased gradient estimates exist in literature, and one key related aspect is the reduction of the variance for these estimates because it affects the stability and speed of convergence of the estimation algorithm. In this PhD thesis we employ the so-called *reparameterization trick* (Kingma and Welling, 2014; Rezende *et al.*, 2014) to effectively do this. When applied to GVA, its essential idea resides on noticing that if there exists an application $\boldsymbol{\theta} = z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})$ such that $\mathfrak{q}(\boldsymbol{\zeta})$ is the $\mathrm{N}(\mathbf{0}, \boldsymbol{I})$ density function $\phi(\boldsymbol{\zeta})$, then:

$$\begin{aligned}
&\nabla_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})) \\
&\quad = E_{\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})} \left\{ \nabla_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta}) - \nabla_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}) \right\} \\
&\quad = E_{\phi(\boldsymbol{\zeta})} \left\{ \nabla_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \mathfrak{p}(\boldsymbol{y}, z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})) - \nabla_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \mathfrak{q}(z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}); \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}) \right\}.
\end{aligned} \tag{1.23}$$

Now the expectation with respect to the standard multivariate Normal distribution can be estimated unbiasedly with samples from it. Moreover, the variational parameters have been moved inside the model joint density function so that the differentiation is done using derivative information from the target posterior density function. In practice, it is found that when the reparameterization trick can be applied it helps greatly to reduce the variance of gradient estimates.

Moreover Roeder *et al.* (2017), generalizing a method described in Han *et al.* (2016) and Tan and Nott (2018), considered a further approach to variance reduction

showing that the last row of (1.23) can be rewritten as:

$$\begin{aligned}&\nabla_{\eta_{q(\theta)}} \log \underline{p}(y; q(\theta; \eta_{q(\theta)})) \\ &= E_{\phi(\zeta)}\left\{ \frac{dz(\zeta;\eta_{q(\theta)})}{d\eta_{q(\theta)}}^T \left(\nabla_\theta \log p(y, z(\zeta; \eta_{q(\theta)})) - \nabla_\theta \log q(z(\zeta; \eta_{q(\theta)}); \eta_{q(\theta)}) \right) \right\},\end{aligned} \tag{1.24}$$

where $dz(\zeta; \eta_{q(\theta)})/d\eta_{q(\theta)}$ is defined as the matrix with (i, j)th element the partial derivative of the ith element of $z(\zeta; \eta_{q(\theta)})$ with respect to the jth element of $\eta_{q(\theta)}$. This latter expression has to be preferred because, if the variational approximation is exact, then a Monte Carlo estimation to the expectation on its right-hand side is exactly zero even if such an estimation is formed using only a single draw from the standardized multivariate Gaussian distribution. Xu *et al.* (2019) showed explicitly how the reparameterization trick reduces the variance of the gradient estimates when $\Sigma_{q(\theta)}$ is assumed to be diagonal.

1.4.3 Parametrizing the Variational Covariance Matrix

Regardless of the strategy employed for solving (1.21), the optimization is well known to be computationally challenging in high-dimensional models because the number of variational parameters to be determined grows as $\mathcal{O}(k^2)$. This makes GVA impractical unless more parsimonious parameterizations of the covariance matrix $\Sigma_{q(\theta)}$ are adopted, and many strategies to overcome this issue have been proposed. For example, Titsias and Lázaro-Gredilla (2014) and Kucukelbir *et al.* (2017) considered both full and diagonal covariance structures expressed in terms of Cholesky factors. In particular, the promising approach of Kucukelbir *et al.* (2017), called *ADVI*, have been recently supported by Stan: the user only provides a probabilistic model and a dataset, and ADVI automatically derives an efficient GVA algorithm making use of the aforementioned techniques. Nonetheless, their full-rank specification still embeds a number of variational parameter which grows quadratically in k. Salimans and Knowles (2013) parametrized the generic approximating Gaussian density function in terms of the precision matrix and exploited sparsity of Hessian matrices for the joint model in their computations. Instead, Tan and Nott (2018) parametrized the variational optimization directly in terms of the Cholesky factor of the precision matrix and imposed sparsity on this factor, reflecting conditional independence relationships. Their approach have been extended by Tan *et al.*

(2020) accounting for a conditionally-structured specification of the generic approximating density function in terms of global and local model parameters. Nevertheless, all these methods either require some special structure of the model or are inflexible in the kinds of dependence they can represent, and in general, do not scale well with high model dimensions.

Alternative approaches rely on parameterizations of $\boldsymbol{\Sigma}_{q(\boldsymbol{\theta})}$ in terms of a factor structure (Bartholomew *et al.*, 2011), allowing for the total number of variational parameters to be reduced considerably when the number of factors considered is much less than the full dimension of the parameter space. Early contributions include Barber and Bishop (1998) and Seeger (2000) but are restricted to models in which the lower bound expression can be evaluated analytically, while Rezende *et al.* (2014) only considered a factor structure for $\boldsymbol{\Sigma}^{-1}_{q(\boldsymbol{\theta})}$ restricted to only one factor component. In this PhD thesis, we opt for a more general approach developed by Ong *et al.* (2018) in which a latent factor structure was considered for the variational covariance matrix, making the total number of variational parameter growing as $\mathcal{O}(kp)$, p being the number of latent factors considered. Such a covariance matrix parameterization may be particularly useful in situations where there is no natural conditional independence structure to be exploited in the model for reducing the number of covariance parameters. GVA approximations with this convenient parameterization are employed in Chapter 4. We finally mention the prominent work of Quiroz *et al.* (2020) in which they combined factor parameterizations for state reduction with sparse precision Cholesky factors for capturing dynamic dependence structure in high-dimensional state space models.

1.5 Approximation Assessment

Once the optimal approximating density $q^*(\boldsymbol{\theta})$ has been obtained, it is of interest to evaluate how good the approximation is if compared to the *exact* posterior density function. Goodness can be measured in many different ways, regardless of the variational approximate method employed: in this PhD thesis, we usually focus on determining how accurate the approximation is and on quantifying the saving in computational runtime if compared to classical MCMC sampling procedures. Issues related to these two major benchmarks are discussed hereafter.

1.5.1 The Accuracy Index

Measuring the global accuracy of the approximation for a k-variate approximating density function $\mathfrak{q}^*(\boldsymbol{\theta})$ if compared to $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ is very complicated even for moderate dimensions of $\boldsymbol{\theta}$. Nonetheless, it is always possible to assess this task for univariate dimensions, one model parameter at a time, both graphically or with appropriate score indexes. Procedures of the former type are usually conducted superimposing the associated univariate approximate posterior density function to the binned kernel density function built with MCMC draws from the *true* posterior distribution. Procedures of the latter type are way more useful since they permit to summarize the quality of that superimposition with only one specific score, and therefore allow immediate comparison between multiple approximations, as provided by the accuracy index first introduced in Section 3.2 of Faes *et al.* (2011) and well described in described in Section 8 of Wand *et al.* (2011).

Let then $\theta \in \Theta$ be a generic univariate parameter belonging to the generic model parameter vector $\boldsymbol{\theta}$, and measure the l_1-distance between $\mathfrak{q}^*(\theta)$ and $\mathfrak{p}(\theta|\boldsymbol{y})$ with the integrated absolute error (IAE) index given by:

$$\text{IAE}\{\mathfrak{q}^*(\theta)\} = \int_{\Theta} |\mathfrak{q}^*(\theta) - \mathfrak{p}(\theta|\boldsymbol{y})| \, \mathrm{d}\theta.$$

Such error measurement has the attractions of being invariant to monotone transformations on θ and to return a scale-independent number between 0 and 2 (Devroye and Györfi, 1985). The second of these characteristics motivates measuring the accuracy of the approximation with the *accuracy index* being defined as:

$$\text{Accuracy}\{\mathfrak{q}^*(\theta)\} = \left(1 - \frac{1}{2}\,\text{IAE}\{\mathfrak{q}^*(\theta)\}\right) \times 100\%, \tag{1.25}$$

taking values from 0% to 100%, with larger scores to be preferred.

Despite its elegant expression, computation of the accuracy index is a little challenging because it depends on the marginal posterior density function $\mathfrak{p}(\theta|\boldsymbol{y})$, which is the element we are trying to avoid by using approximate variational methods. Nonetheless, sufficiently large samples obtained from MCMC can be used to adequately estimate it employing binned kernel density estimation with direct plug-in bandwidth selection (see e.g. Section 3.6.1 of Wand and Jones, 1995), as facilitated by the R package KernSmooth (Wand, 2020). The integration in (1.25) is then accurately solved via numerical quadrature.

Notice the obvious choice of the Kullback-Leibler divergence as an alternative to

(1.25) has to be avoided here since it can be dominated by the tail behavior of the densities involved, as pointed out by Hall (1987).

1.5.2 Runtime Comparisons

Another critical aspect for benchmarking variational approximations is the computational runtime required for obtaining the optimal $\mathfrak{q}^*(\boldsymbol{\theta})$ density function. Regardless of the approximating method employed, obtaining a reliable measurement is typically out of the question because it would be conditioned by many technical details involved in the algorithm implementation. Some of them include the efficiency of the updating scheme to be manually programmed in standard statistical software, the way convergence was assessed and its associated threshold value, the programming language and data structures employed, possible fixed data manipulations or iteration-invariant matrix multiplications being performed prior running the estimating algorithm. Therefore only runtime comparisons between different procedures keeping all these aspects fixed are possible and scientifically significant.

Variational approximations are usually compared to MCMC sampling strategies even in their respective runtime. However, even in this case, it is challenging to quantify the speed gains scientifically due to many different options making the different algorithms incomparable. For example, the global MCMC sampling runtime is conditioned by the number of iterations required to claim the convergence of the underlying Markov chain to the stationary distribution, by the sampling strategy, by the programming language on which it is implemented (including the usage of probabilistic languages such as Stan or Nimble) and whether possible sparse matrix structures have been exploited. Detailed discussions on this topic are given in, e.g., Section 5.1 of Ormerod *et al.* (2017) and Section 7 of Nolan *et al.* (2020).

Chapter 2

Power-EP Approximations Based on the α-Divergence

2.1 Introduction

The primary goal of this chapter is to investigate a class of approximations based on a divergence measure generalizing the Kullback-Leibler divergence, called α-divergence, for some of the most popular statistical models. As shown in Chapter 1, all the most common variational approximation methods are based upon the Kullback-Leibler divergence quantifying the *distance* between $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ and the generic $\mathfrak{q}(\boldsymbol{\theta}) \in \mathcal{Q}$. This choice is usually motivated by the tractability of the resulting associated optimization scheme; nonetheless, a generic variational approximation (1.1) is defined for a divergence measure $\mathcal{D}$ not necessarily corresponding to the Kullback-Leibler divergence. Therefore, many competing variational approximation methods have been developed and proposed in recent years, often exhibiting better approximation performances or possessing more reliable convergence guarantees. A recent overview of popular divergence measures employed in statistics is Nielsen (2020), to which we refer.

The literature on variational approximations with divergences alternative to the Kullback-Leibler divergence is vast and grows year by year. For example, Li and Turner (2016) solved (1.1) with $\mathcal{D}$ being a member of the family of Rényi divergences of order α, while Dieng *et al.* (2017) minimized $\mathcal{D}_{\chi^2}\{\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}\|\mathfrak{q}(\boldsymbol{\theta})\}$, $\mathcal{D}_{\chi^2}$ representing the χ^2-divergence. Liu and Wang (2016) connected the Kullback-Leibler divergence with a recently proposed kernelized Stein divergence which is indeed minimized. Recent promising work is due to Knoblauch *et al.* (2019), in which they defined a completely new class of generalized variational inference methods, allowing for a

wide family of divergence measures to be minimized. Saha *et al.* (2020) proposed a novel Riemannian geometric framework for variational inference based on the nonparametric Fisher-Rao metric on the manifold of probability density functions, reformulating the task of approximating the posterior distribution as a variational problem on the hypersphere based on the Rényi divergences of order α. Huggins *et al.* (2020) discussed alternative criteria, including the notable Wasserstein divergence, and provided some theoretical guarantees. A complete review of Bayesian approximate methods goes beyond the scope of this PhD thesis, and Section 5.2 of Zhang *et al.* (2019) contain additional references to this topic.

In this chapter, we focus on variational approximations minimizing a member of the α-divergences family (Amari, 1985), expressible with the convention of Zhu and Rohwer (1997) as:

$$\mathcal{D}_\alpha\{\mathfrak{p}(\boldsymbol{x})\|\mathfrak{q}(\boldsymbol{x})\} = \frac{1}{\alpha(1-\alpha)}\left\{1 - \int_{\mathcal{X}} \mathfrak{p}(\boldsymbol{x})^\alpha \mathfrak{q}(\boldsymbol{x})^{1-\alpha}\,\mathrm{d}\boldsymbol{x}\right\}, \quad \alpha \in \mathbb{R}\backslash\{0\}, \tag{2.1}$$

for two generic probability density functions $\mathfrak{p}(\boldsymbol{x})$ and $\mathfrak{q}(\boldsymbol{x})$ having support in $\mathcal{X}$. Although having similar name and notation, and sharing connections as pointed out in Section 2.2 of Cichocki and Amari (2010), (2.1) must not be confounded with the popular Rényi-divergences of order α (Rényi, 1961), having expression:

$$\mathcal{D}_\alpha^{\text{Rényi}}\{\mathfrak{p}(\boldsymbol{x})\|\mathfrak{q}(\boldsymbol{x})\} = \frac{1}{\alpha-1}\log\left(\int_{\mathcal{X}} \mathfrak{p}(\boldsymbol{x})^\alpha \mathfrak{q}(\boldsymbol{x})^{1-\alpha}\,\mathrm{d}\boldsymbol{x}\right), \quad \alpha \in \mathbb{R}\backslash\{1\}.$$

Additional relations of the α-divergence with the f-divergence (Ali and Silvey, 1966) and the Bregman divergence (Bregman, 1967) classes have been pointed out by Amari (2009). As it is clear by its definition, a divergence measure is selected among this family only once a specific value for α is chosen, and we emphasize this fact adding a subscript on $\mathcal{D}_\alpha$.

Solving (1.1) with the α-divergence (2.1) chosen as reversed divergence measure defines *Power-Expectation Propagation (Power-EP)*. It represents an extension of EP proposed by Minka (2004) for solving some of its well-known drawbacks, e.g., the convergence which is not always guaranteed (see Minka, 2001a), and possibly obtaining better approximations. A message-passing algorithm for solving the underlying minimization problem has been proposed by Minka (2005). It generalizes the EP message-passing algorithm summarized in Chapter 1, which turns out to be equivalent to local Kullback-Leibler divergence minimizations of the exact distributions raised to a power α (Minka, 2004). This also motivates the name *Power-EP*.

Among infinitely many choices for α, we highlight two major notable cases of interest for which (2.1) gained attractiveness over the years and connect with variational methods described in Chapter 1. If $\alpha = 1$, then

$$\mathcal{D}_1\{p(x)\|q(x)\} = \mathcal{KL}\{p(x)\|q(x)\}$$

while if $\alpha \to 0$, then

$$\mathcal{D}_{\to 0}\{p(x)\|q(x)\} = \mathcal{KL}\{q(x)\|p(x)\}.$$

Therefore, Power-EP unifies both EP approximations and VB approximations under the same umbrella, because once a unique method for solving its associated minimization problem is defined regardless of the value chosen for α, then both arise as particular cases only selecting $\alpha = 1$ and $\alpha \to 0$, respectively.

These appealing characteristics drove our interest in exploring Power-EP applications to some of the most common statistical models. Rather than considering VB and EP approximations as two opposite and competing approaches, as it usually happens in standard variational literature, we recast them as boundaries of an infinitely uncountable set of approximations that arise within a specific choice for $\alpha \in (0, 1]$. We restrict our attention to all the variational approximations conceptually lying *in between* VB and EP and investigate their behavior for different values of α. In principle, there could be some model-specific values of $\alpha \in (0, 1]$ that share the approximating behavior of VB and EP, allowing for optimal and more accurate approximations between the two.

Little attention has been paid to developing explicit algorithms that account for this type of approximations and statistical applications are still mostly unexplored, further motivating our research interest towards this direction. For each statistical model considered, we present a detailed and explicit Power-EP message-passing algorithm for obtaining the optimal approximating posterior density functions, highlighting benefits and possible drawbacks connected to their implementations. Interestingly, the algorithms we propose derive Power-EP approximations following the same structure on the iterative updating steps, regardless of the approximation required. What changes between one approximation and another is the choice of $\alpha \in (0, 1]$ before initializing the estimating algorithm.

Our work shares common points with Hernandez-Lobato *et al.* (2016). However, they solved the underlying optimization problem using a stochastic gradient descent algorithm and did not exploit the message-passing formulation of the underlying

fixed-point iteration scheme. Although Power-EP approximations are still somehow overshadowed by standard EP approximations, recent works in which they are proved to exhibit better performances include Jylänki *et al.* (2011), Bui *et al.* (2016), Bui *et al.* (2017) and Wilkinson *et al.* (2020).

2.2 Power Expectation Propagation

Due to the strict connection of Power-EP with EP, its mathematical treatment (Minka, 2005) follows that for EP summarized in Section 1.3 with minor changes. Given a model with posterior density function $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$, Power-EP approximations result from the following optimization problem:

$$\mathfrak{q}^*_\alpha(\boldsymbol{\theta}) = \underset{\mathfrak{q}(\boldsymbol{\theta})\in\mathcal{Q}}{\arg\min}\, \mathcal{D}_\alpha\{\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})\|\mathfrak{q}(\boldsymbol{\theta})\}, \quad \alpha \in \mathbb{R}\backslash\{0\}. \tag{2.2}$$

A subscript α is added on the optimal approximating density function to highlight that it solves (2.2) within a specific α-divergence indexed by a pre-specified α value, as different α-divergences may lead to possible different and competing solutions.

Solution of (2.2) can be obtained with different approaches. Coherently with the formulation summarized for EP in Section 1.3, we adopt a message-passing algorithm for solving (2.2) over a generic factor graph representation for a Bayesian model that follows from a pre-specified partition of $\boldsymbol{\theta}$ into $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$. It iteratively computes and updates, for each non-trivial combination of $1 \leq i \leq M$ and $1 \leq j \leq N$, the *Power-EP stochastic node to factor message*:

$$\mathfrak{m}^{(\alpha)}_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i) \longleftarrow \prod_{j' \neq j \,:\, i \in \text{neigh}(j')} \mathfrak{m}^{(\alpha)}_{f_{j'} \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \tag{2.3}$$

and the *Power-EP factor to stochastic node message:*

$$\mathfrak{m}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i) \longleftarrow \frac{\text{proj}_{\mathcal{Q}_i}\left[\begin{array}{l} Z^{-1}\left(\mathfrak{m}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i) \int \left[f_j(\boldsymbol{\theta}_{\text{neigh}(j)})\right]^\alpha \\ \times \prod_{i' \in \text{neigh}(j)\backslash\{i\}} \left(\mathfrak{m}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_{i'}}(\boldsymbol{\theta}_{i'})\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\boldsymbol{\theta}_{i'} \to f_j}(\boldsymbol{\theta}_{i'})\, \mathrm{d}\boldsymbol{\theta}_{\text{neigh}(j)\backslash\{i\}} \end{array}\right]}{\mathfrak{m}^{(\alpha)}_{\boldsymbol{\theta}_i \to f_j}(\boldsymbol{\theta}_i)}. \tag{2.4}$$

Upon convergence of the messages, the optimal $\mathfrak{q}_\alpha$-densities are obtained with:

$$\mathfrak{q}^*_{i,\alpha}(\boldsymbol{\theta}_i) \propto \prod_{j : i \in \text{neigh}(j)} \mathfrak{m}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i), \quad 1 \leq i \leq M. \tag{2.5}$$

Notice the right-hand side of (2.3) corresponds to the one of (1.9) and, similarly, the right-hand side of (2.5) corresponds to the one of (1.14). The reason is due to the structure of the message-passing algorithm, which is always the same for any type of variational approximation required. Moreover, the Power-EP lower bound $\log \underline{\mathrm{p}}(\boldsymbol{y}; \mathrm{q}_\alpha(\boldsymbol{\theta}))$ is expressible as:

$$\log \underline{\mathrm{p}}(\boldsymbol{y}; \mathrm{q}_\alpha(\boldsymbol{\theta})) = \sum_{i=1}^{M} \log s_{\boldsymbol{\theta}_i}^{(\alpha)} + \frac{1}{\alpha} \sum_{j=1}^{N} \log s_{f_j}^{(\alpha)} \tag{2.6}$$

where

$$s_{\boldsymbol{\theta}_i}^{(\alpha)} = \int \prod_{j: i \in \mathrm{neigh}(j)} \mathrm{m}_{f_j \to \boldsymbol{\theta}_i}^{(\alpha)}(\boldsymbol{\theta}_i) \, \mathrm{d}\boldsymbol{\theta}_i \tag{2.7}$$

and

$$s_{f_j}^{(\alpha)} = \frac{\int \left[f_j(\boldsymbol{\theta}_{\mathrm{neigh}(j)}) \right]^{\alpha} \prod_{i \in \mathrm{neigh}(j)} \left(\mathrm{m}_{f_j \to \boldsymbol{\theta}_i}^{(\alpha)}(\boldsymbol{\theta}_i) \right)^{1-\alpha} \mathrm{m}_{\boldsymbol{\theta}_i \to f_j}^{(\alpha)}(\boldsymbol{\theta}_i) \, \mathrm{d}\boldsymbol{\theta}_{\mathrm{neigh}(j)}}{\int \prod_{i \in \mathrm{neigh}(j)} \mathrm{m}_{\boldsymbol{\theta}_i \to f_j}^{(\alpha)}(\boldsymbol{\theta}_i) \, \mathrm{m}_{f_j \to \boldsymbol{\theta}_i}^{(\alpha)}(\boldsymbol{\theta}_i) \, \mathrm{d}\boldsymbol{\theta}_{\mathrm{neigh}(j)}}. \tag{2.8}$$

If all the M different $\mathrm{q}_i(\boldsymbol{\theta}_i) \in \mathcal{Q}_i$ such that $\mathrm{q}(\boldsymbol{\theta}) = \prod_{i=1}^{M} \mathrm{q}_i(\boldsymbol{\theta}_i)$ are selected among the exponential family of distributions, as it usually happens for practical EP and Power-EP implementations, then the Power-EP message-passing expressions between a generic factor f and stochastic node $\boldsymbol{\theta}$ become

$$\mathrm{m}_{f \to \boldsymbol{\theta}}^{(\alpha)}(\boldsymbol{\theta}) \propto \exp\left\{ \boldsymbol{T}(\boldsymbol{\theta})^T \boldsymbol{\eta}_{f \to \boldsymbol{\theta}}^{(\alpha)} - A\left(\boldsymbol{\eta}_{f \to \boldsymbol{\theta}}^{(\alpha)} \right) \right\},$$

$$\mathrm{m}_{\boldsymbol{\theta} \to f}^{(\alpha)}(\boldsymbol{\theta}) \propto \exp\left\{ \boldsymbol{T}(\boldsymbol{\theta})^T \boldsymbol{\eta}_{\boldsymbol{\theta} \to f}^{(\alpha)} - A\left(\boldsymbol{\eta}_{\boldsymbol{\theta} \to f}^{(\alpha)} \right) \right\}$$

and the message-passing updates boil down to the solely updating of the associated natural parameter vectors. In particular, update (2.3) reduces into

$$\boldsymbol{\eta}_{\boldsymbol{\theta}_i \to f_j}^{(\alpha)} \longleftarrow \sum_{j' \neq j \,:\, i \in \mathrm{neigh}(j')} \boldsymbol{\eta}_{f_{j'} \to \boldsymbol{\theta}_i}^{(\alpha)}, \tag{2.9}$$

since the vector of sufficient statistics $\boldsymbol{T}(\boldsymbol{\theta}_i)$ is the same for each neighboring factor f_j, and update (2.4) reduces into

$$\boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i}^{(\alpha)} \longleftarrow \boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i}^{*(\alpha)} - \boldsymbol{\eta}_{\boldsymbol{\theta}_i \to f_j}^{(\alpha)}, \tag{2.10}$$

with $\boldsymbol{\eta}_{f_j \to \boldsymbol{\theta}_i}^{*(\alpha)}$ indicating the optimal natural parameter vector (1.13) resulting from the

Algorithm 2.1 *General implementation of a Power-EP message-passing algorithm for solving* (2.2) *on a Bayesian statistical model representable as a factor graph following the prespecified parameter partition* $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$.

1. Select $\alpha \in (0, 1]$.
2. Initialize the natural parameter vectors $\boldsymbol{\eta}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}$ of the Power-EP factor to stochastic node messages belonging to a suitably-chosen exponential family of distributions $\mathcal{Q}_i$, for all the non-trivial combinations of $1 \leq i \leq M$ and $1 \leq j \leq N$.
3. Until convergence, cycle:

   For each factor f_j, $1 \leq j \leq N$:

   3.1 Update all the natural parameter vectors $\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\theta}_i \to f_j}$ of the Power-EP stochastic node to factor messages using (2.9), for all the relevant $1 \leq i \leq M$;

   3.2 Update all the natural parameter vectors $\boldsymbol{\eta}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}$ of the Power-EP factor to stochastic node messages using (2.10), for all the relevant $1 \leq i \leq M$;
4. Compute and return the natural parameter vectors $\boldsymbol{\eta}_{\mathrm{q}^*_{i,\alpha}(\boldsymbol{\theta}_i)}$ corresponding to the optimal approximating densities using (2.11), for all $1 \leq i \leq M$.

Kullback-Leibler projection into $\mathcal{Q}_i$. Upon convergence of all the natural parameter vector updates, the optimal q_α-densities (2.5) are uniquely identified by their natural parameter vectors to be computed as:

$$\boldsymbol{\eta}_{\mathrm{q}^*_{i,\alpha}(\boldsymbol{\theta}_i)} = \sum_{j: i \in \mathrm{neigh}(j)} \boldsymbol{\eta}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}, \quad 1 \leq i \leq M. \tag{2.11}$$

For all the messages, natural parameter vectors and associated quantities, we stress that they depend upon the minimization of a specific α-divergence adding the superscript (α). All the expressions for EP presented in Section 1.3 arise from those presented in the previous pages fixing $\alpha = 1$: Power-EP generalizes and includes it among a broad set of competing approximations indexed by α. In this chapter we focus on $\alpha \in (0, 1]$ values being those conceptually *bridging the gap* between VB and EP. For the remainder of this chapter, we also introduce the following notation:

$$\boldsymbol{\eta}^{(\alpha)}_{f_j \leftrightarrow \boldsymbol{\theta}_i} \equiv (1 - \alpha) \boldsymbol{\eta}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i} + \boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\theta}_i \to f_j}.$$

Algorithm 2.1 gives a general structure for the Power-EP message-passing algorithm. The only step requiring a considerable computational effort is Step 3.2, because it needs to perform the required Kullback-Leibler projection into $\mathcal{Q}_i$. All the other steps are straightforward manipulations of natural parameter vectors.

2.3 Power-EP for the Univariate Normal Random Sample Model

Dealing with the concepts of *factor graphs*, *Kullback-Leibler projections* and the notation for the natural parameter vector update expressions defined in the previous pages can be quite complicated. Therefore, difficulties in correctly interpreting how the Power-EP message-passing algorithm works are frequent. For this reason, inspired by the excellent work of Kim and Wand (2016), we describe in detail Power-EP approximations for the simplest statistical model ever, the univariate Normal random sample model, and explicitly derive the required algebra for computing the associated natural parameter vector updates.

Assume then a random sample $\boldsymbol{y}$ of length n has been generated from a univariate Normal distribution, and consider the following model specification:

$$\begin{aligned} y_i|\mu,\sigma^2 &\overset{\text{iid}}{\sim} \mathrm{N}(\mu,\sigma^2) \quad \text{for} \quad 1 \leq i \leq n, \\ \mu &\sim \mathrm{N}(\mu_\mu,\sigma_\mu^2), \quad \sigma \sim \text{Half-Cauchy}(s) \end{aligned}$$

where $\mu_\mu \in \mathbb{R}$, $\sigma_\mu^2, s > 0$ are prior hyperparameters controlling the prior beliefs about the mean parameter μ and variance parameter σ^2. Following Gelman (2006), we set the Half-Cauchy prior distribution for the standard deviation parameter σ, which usually better lends itself to weakly-uninformative prior specification. A useful result for Half-Cauchy distributions, as recalled in Appendix A.2, helps restore to the popular Inverse-Gamma (Inverse-χ^2) conjugate prior specification for the variance parameter σ^2, at the cost of introducing the augmented variable a. Therefore, an equivalent Bayesian formulation is:

$$\begin{aligned} \boldsymbol{y}|\mu,\sigma^2 &\sim \mathrm{N}(\mu\mathbf{1},\sigma^2\boldsymbol{I}), \\ \mu \sim \mathrm{N}(\mu_\mu,\sigma_\mu^2), \quad \sigma^2|a &\sim \text{Inverse-Gamma}(1/2,1/a), \\ a &\sim \text{Inverse-Gamma}(1/2,1/s^2). \end{aligned} \tag{2.12}$$

Here $\boldsymbol{\theta} = (\mu,\sigma^2,a)^T$ and we partition it into $M = 3$ stochastic nodes, namely $\boldsymbol{\theta}_1 = \{\mu\}$, $\boldsymbol{\theta}_2 = \{\sigma^2\}$ and $\boldsymbol{\theta}_3 = \{a\}$. Such a partitioning choice has a crucial impact on the quality of the Power-EP approximation, as it implies the following factorization for the approximating density function: $\mathfrak{q}(\mu,\sigma^2,a) = \mathfrak{q}(\mu)\,\mathfrak{q}(\sigma^2)\,\mathfrak{q}(a)$. Moreover, the joint density function $\mathfrak{p}(\boldsymbol{y},\mu,\sigma^2,a) = \mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\,\mathfrak{p}(\sigma^2|a)\,\mathfrak{p}(\mu)\,\mathfrak{p}(a)$ naturally factorizes itself into $N = 4$ distinct factors.

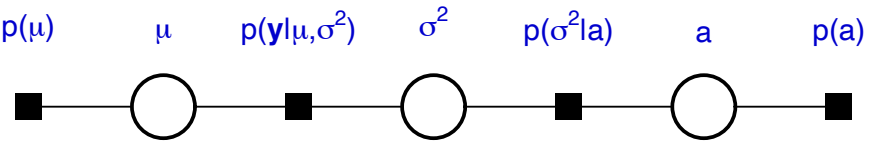

FIGURE 2.1: *Factor graph representation of model* (2.12) *following parameter partition* $\{\{\mu\},\{\sigma^2\},\{a\}\}$ *and joint density factorization* $\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\,\mathfrak{p}(\sigma^2|a)\,\mathfrak{p}(\mu)\,\mathfrak{p}(a)$.

Figure 2.1 shows the factor graph representation of model (2.12) with the pre-specified global parameter partition. Each of the $M = 3$ stochastic nodes is represented with an empty circle, while each of the $N = 4$ factors is represented with a full squared box and corresponds to each of the factors into which $p(\boldsymbol{y},\mu,\sigma^2,a)$ is factorized accordingly to the pre-specified partition of $(\mu,\sigma^2,a)^T$. An edge, represented with a solid straight line, connects all the stochastic nodes belonging to the neighborhood of each factor, since $\text{neigh}(\mathfrak{p}(\mu)) = \{\mu\}$, $\text{neigh}(\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)) = \{\mu,\sigma^2\}$, $\text{neigh}(\mathfrak{p}(\sigma^2|a)) = \{\sigma^2,a\}$ and $\text{neigh}(\mathfrak{p}(a)) = \{a\}$.

Power-EP then solves the following optimization problem:

$$\mathfrak{q}^*_\alpha(\mu,\sigma^2,a) = \underset{\mathfrak{q}(\mu,\sigma^2,a)\in\mathcal{Q}}{\arg\min}\ \mathcal{D}_\alpha\{\mathfrak{p}(\mu,\sigma^2,a|\boldsymbol{y})\|\mathfrak{q}(\mu,\sigma^2,a)\},\quad \alpha\in\mathbb{R}\backslash\{0\}, \tag{2.13}$$

with $\mathcal{Q} = \{\mathfrak{q}(\mu,\sigma^2,a) : \mathfrak{q}(\mu,\sigma^2,a) = \mathfrak{q}(\mu)\mathfrak{q}(\sigma^2)\mathfrak{q}(a)\}$ and the following convenient exponential family distributions selected for each approximating density function:

$$\begin{aligned}
&\mathfrak{q}(\mu) \text{ is a } \mathrm{N}\big(\mu_{\mathfrak{q}(\mu)},\sigma^2_{\mathfrak{q}(\mu)}\big) \text{ density function ,}\\
&\mathfrak{q}(\sigma^2) \text{ is an Inverse-Gamma}\big(\xi_{\mathfrak{q}(\sigma^2)},\lambda_{\mathfrak{q}(\sigma^2)}\big) \text{ density function ,}\\
&\mathfrak{q}(a) \text{ is an Inverse-Gamma}\big(\xi_{\mathfrak{q}(a)},\lambda_{\mathfrak{q}(a)}\big) \text{ density function .}
\end{aligned} \tag{2.14}$$

Notice these generic approximating $\mathfrak{q}$-densities are not equipped with the α subscript in their notation, because the generic family $\mathcal{Q}$ is the same regardless of the α-divergence chosen in (2.13). Conversely, $\mathfrak{q}^*_\alpha(\mu)$, $\mathfrak{q}^*_\alpha(\sigma^2)$ and $\mathfrak{q}^*_\alpha(a)$ are the associated optimal solutions minimizing a pre-specified α-divergence, uniquely identified by their associated natural parameter vectors $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\mu)}$, $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)}$ and $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)}$, and therefore exhibit the α subscript.

Choosing $\mathcal{Q}_\mu$, $\mathcal{Q}_{\sigma^2}$ and $\mathcal{Q}_a$ to belong to the exponential family of distributions boils down the message-passing algorithm into the only need of updating the natural parameter vectors of the Power-EP stochastic node to factor messages (2.9),

having explicit expressions:

$$\begin{gathered}
\boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\mu)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}, \quad \boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu}, \\
\boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}, \quad \boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\sigma^2|a)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}, \\
\boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(\sigma^2|a)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}, \quad \boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(a)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a},
\end{gathered} \tag{2.15}$$

and the natural parameter vectors of the Power-EP factor to stochastic node messages (2.10), having explicit expressions:

$$\begin{gathered}
\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\mu)\to\mu} - \boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\mu)}, \\
\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu} - \boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}, \\
\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} - \boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}, \\
\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} - \boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\sigma^2|a)}, \\
\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} - \boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(\sigma^2|a)}, \\
\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a} - \boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(a)}.
\end{gathered} \tag{2.16}$$

Once the iterative refinements of the natural parameter vectors have reached convergence, application of (2.11) gives the natural parameter vectors of the Power-EP optimal approximating $\mathfrak{q}$-densities, namely:

$$\begin{aligned}
\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\mu)} &= \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}, \\
\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)} &= \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}, \\
\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)} &= \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}
\end{aligned} \tag{2.17}$$

and therefore the optimal $\mathfrak{q}$-densities are:

$$\begin{aligned}
&\mathfrak{q}^*_\alpha(\mu) \text{ is the } \mathrm{N}\big(-[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\mu)}]_1/(2[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\mu)}]_2), -1/(2[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\mu)}]_2)\big) \text{ density function}, \\
&\mathfrak{q}^*_\alpha(\sigma^2) \text{ is the Inverse-Gamma}\big(-[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)}]_1 - 1, -[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)}]_2\big) \text{ density function}, \\
&\mathfrak{q}^*_\alpha(a) \text{ is the Inverse-Gamma}\big(-[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)}]_1 - 1, -[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)}]_2\big) \text{ density function}.
\end{aligned} \tag{2.18}$$

The only step requiring a non-negligible computational effort concerns the computation of the natural parameter vectors $\boldsymbol{\eta}^{*(\alpha)}_{f_j\to\boldsymbol{\theta}_i}$ with (1.13), for each non-trivial combination of $1 \leq i \leq M$ and $1 \leq j \leq N$. With direct reference to model (2.12),

this means computing $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$ performing a Kullback-Leibler projection onto the univariate Normal exponential family of distributions, and computing $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a}$ performing a Kullback-Leibler projection onto the Inverse-Gamma (Inverse-χ^2) exponential family of distributions. The following two results derived by Kim and Wand (2016) give an operational strategy for determining such natural parameter vectors.

Result 2.1. *Let X be a non-degenerate random variable for which $E(X^2)$ exists and with density function $\mathfrak{p}(x)$. The Kullback-Leibler projection of $\mathfrak{p}(x)$ onto the univariate Normal family,*

$$\mathfrak{q}_{KL}(x;\mu_{KL},\sigma^2_{KL}) = proj_N[\mathfrak{p}(x)] = \underset{\mathfrak{q}(x;\mu,\sigma^2)\in N(\mu,\sigma^2)}{\arg\min} \mathcal{KL}\{\mathfrak{p}(x)\|\mathfrak{q}(x;\mu,\sigma^2)\}.$$

is the $N(\mu_{KL},\sigma^2_{KL})$ density function with mean and variance parameters

$$\mu_{KL} = E(X) \quad \textit{and} \quad \sigma^2_{KL} = E(X^2) - (\mu_{KL})^2,$$

respectively.

Result 2.2. *Let X be a positive-valued non-degenerate random variable for which $E(1/X)$ and $E(\log X)$ exist and with density function $\mathfrak{p}(x)$. The Kullback-Leibler projection of $\mathfrak{p}(x)$ onto the Inverse Gamma family,*

$$\mathfrak{q}_{KL}(x;\xi_{KL},\lambda_{KL}) = proj_{IG}[\mathfrak{p}(x)] = \underset{\mathfrak{q}(x;\xi,\lambda)\in IG(\xi,\lambda)}{\arg\min} \mathcal{KL}\{\mathfrak{p}(x)\|\mathfrak{q}(x;\xi,\lambda)\}.$$

is the Inverse-Gamma(ξ_{KL},λ_{KL}) density function with shape and scale parameters

$$\xi_{KL} = (log\text{–}\psi)^{-1}\{\log E(1/X) + E(\log X)\} \quad \textit{and} \quad \lambda_{KL} = \xi_{KL}/E(1/X),$$

respectively. Here $(log\text{–}\psi)(x) \equiv \log(x) - \psi(x)$ is defined for all $x > 0$ and can be effectively computed in R with the logmdigamma() function in the statmod library (Smyth et al.*, 2021). Theorem 1 of Kim and Wand (2016) proves it is a proper bijective mapping from $\mathbb{R}^+$ to $\mathbb{R}^+$ and gives further technicalities on how to efficiently compute its inverse mapping.*

Algorithm 2.2 gives a synthetic and general scheme for performing Power-EP approximations on model (2.12) solving (2.13) with approximating density functions restricted to the parametric family described in (2.14). Without loss of generality, we initialize $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$ to be the natural parameter vector of a $\mathrm{N}(0,1)$ distribution,

Algorithm 2.2 *Power Expectation Propagation message-passing algorithm on factor graph displayed in Figure 2.1 for determining the natural parameter vectors of the optimal density functions* (2.18) *for approximate Bayesian inference on model* (2.12).

Data Inputs: $\boldsymbol{y}$ $(n \times 1)$. Create summary data vector $\boldsymbol{D} = (n, \boldsymbol{y}^T\mathbf{1}, \|\boldsymbol{y}\|^2)^T$.

Power-EP α choice: $\alpha \in (0, 1]$.

Hyperparameter Inputs: $\mu_\mu \in \mathbb{R}$, σ_μ and $s > 0$.

Initialize: $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu} \leftarrow \begin{bmatrix} \mu_\mu/\sigma_\mu^2 \\ -1/(2\sigma_\mu^2) \end{bmatrix}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} \leftarrow \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu} \leftarrow \begin{bmatrix} 0 \\ -1/2 \end{bmatrix}$,

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} \leftarrow \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \quad \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \leftarrow \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \quad \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \leftarrow \begin{bmatrix} -2 \\ -1 \end{bmatrix}.$$

$\boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)} \leftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$; $\boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(\sigma^2|a)} \leftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}$.

Cycle until convergence:

Updates for the $\mathfrak{p}(\mu)$ factor: fixed.

Updates for the $\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)$ factor:

$$\boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu} \longleftarrow G^{\mathrm{N}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}; \alpha\boldsymbol{D}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} \longleftarrow G^{\mathrm{IG1}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}; \alpha\boldsymbol{D}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}.$$

Updates for the $\mathfrak{p}(\sigma^2|a)$ factor:

$$\boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\sigma^2|a)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \longleftarrow G^{\mathrm{IG2}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}, \begin{bmatrix} [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}]_1 + 2(1-\alpha) \\ [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}]_2/\alpha \end{bmatrix}; 3\alpha\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \longleftarrow G^{\mathrm{IG2}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}, \begin{bmatrix} [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}]_1 + 2(1-\alpha) \\ [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}]_2/\alpha \end{bmatrix}; \alpha\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}.$$

Updates for the $\mathfrak{p}(a)$ factor: fixed.

Outputs: $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\mu)} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$, $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$,

$\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$.

---

while $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ to be the natural parameter vectors of an Inverse-Gamma$(1, 1)$ distribution. Details on the G^{N}, G^{IG1} and G^{IG2} wrappers and derivations of the associated Kullback-Leibler projected natural parameter vectors are given in Appendices B.1 and B.2, respectively.

Performing approximate Bayesian inference on the univariate Normal random sample model is more a theoretical exercise than a useful inferential strategy. Nonetheless, it gives a deepened illustration of how the Power-EP message-passing algorithm works, allowing for a straightforward interpretation of the generic update expressions.

Two significant considerations emerge. Firstly, Algorithm 2.2 generalizes Algorithm 1 of Kim and Wand (2016) for performing EP on the same model specification, allowing for $\alpha \in (0, 1]$ and potentially for an infinitely uncountable set of approximate distributions to the model posterior density function. This comes just specifying a proper value for α before initializing the algorithm, preserving the convenient structures of the integral-defined functions and underlying numerical routines, together with the natural parameter vector updates expressions. If $\alpha = 1$ is selected, we obtain the same results for their EP approximation methodology. Secondly, having a closer look at how the α value affects Algorithm 2.2, we see that it acts on the evaluation of the Kullback-Leibler projection associated wrappers G^{N} and G^{IG1} defined in Kim and Wand (2016). In fact, it selects a proportion of the summary data vector $\boldsymbol{D}$ in their $\boldsymbol{c}$ argument and a proportion of the Power-EP factor contribution to stochastic node message natural parameter $\boldsymbol{\eta}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i}$ to be included in their $\boldsymbol{a}$ and $\boldsymbol{b}$ arguments (just remember how $\boldsymbol{\eta}^{(\alpha)}_{f_j \leftrightarrow \boldsymbol{\theta}_i}$ is defined). Similar comments apply to the G^{IG2} related updates, although with a less intuitive and clear interpretation.

We emphasize that the most computationally-challenging operation regards the Kullback-Leibler projection onto a suitably-chosen distribution belonging to the exponential family. This comes in the univariate Normal random sample model and similar univariate-like models with unidimensional numerical integration techniques to be performed, for which fast numerical routines usually provide accurate results. In the remainder of this chapter, we propose a novel methodology for performing Power-EP approximations on some regression models. What differs from what has been reported so far is that Kullback-Leibler projections onto the Multivariate Normal family are required to achieve optimal approximation accuracies. Such projections require multivariate numerical integration techniques, which may slow down the estimating process and make the overall Bayesian approximating algorithm useless. Kim and Wand (2018) tackled this problem for EP approximations introducing what they called *derived variables* to circumvent it; nonetheless, this strategy becomes unfeasible in large-data settings and is not amenable for Power-EP extension. The following pages demonstrate how their strategy can be circumvented for developing Power-EP approximations for some common regression models.

2.4 Power-EP for the Normal Linear Regression Model

Let us consider the following linear regression model with Normal responses:

$$\begin{gathered} \boldsymbol{y}|\boldsymbol{\beta},\sigma^2 \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{\beta},\sigma^2\boldsymbol{I}), \\ \boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \quad \sigma^2|a \sim \text{Inverse-Gamma}(1/2,1/a), \\ a \sim \text{Inverse-Gamma}(1/2,1/s^2), \end{gathered} \tag{2.19}$$

where $\boldsymbol{y}$ is the observed response vector $\boldsymbol{y}$ of dimension n, $\boldsymbol{X} = (\boldsymbol{x}_1^T,\ldots,\boldsymbol{x}_n^T)^T$ is the design regressor matrix of dimension $n \times d$, $\boldsymbol{\beta}$ is a d-dimensional vector of regression coefficients and σ^2 is the variance parameter for the unit-specific error term. A multivariate Normal prior distribution for $\boldsymbol{\beta}$ is chosen, with $\boldsymbol{\mu}_{\boldsymbol{\beta}} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ being a symmetric positive definite matrix of dimension $d \times d$. Moreover, the aforementioned Half-Cauchy(s) prior is placed over the standard deviation parameter σ. Notice that if $d = 1$ and $\boldsymbol{X} = \boldsymbol{1}$, then model (2.12) is found; in the following we exclude this degenerate particular case always assuming $d > 1$, i.e. that $\boldsymbol{X}$ contains at least one regressor column in addition to the intercept.

Bayesian inference focuses on determining the posterior density function having expression $\mathfrak{p}(\boldsymbol{\beta},\sigma^2,a|\boldsymbol{y}) \propto \mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\,\mathfrak{p}(\boldsymbol{\beta})\,\mathfrak{p}(\sigma^2|a)\,\mathfrak{p}(a)$, and we aim to find the optimal Power-EP posterior approximating density $\mathfrak{q}^*_\alpha(\boldsymbol{\beta},\sigma^2,a)$ that best solves (2.2) for a fixed $\alpha \in (0,1]$. Differently from the univariate Normal random sample model, here $\boldsymbol{\theta} = (\boldsymbol{\beta}^T,\sigma^2,a)^T$ with $\boldsymbol{\beta}$ being d-dimensional: we consider the partition $\{\{\boldsymbol{\beta}\},\{\sigma^2\},\{a\}\}$ for $\boldsymbol{\theta}$, from which the following factorization $\mathfrak{q}(\boldsymbol{\beta},\sigma^2,a) = \mathfrak{q}(\boldsymbol{\beta})\,\mathfrak{q}(\sigma^2)\,\mathfrak{q}(a)$ for the generic $\mathfrak{q}(\boldsymbol{\theta}) \in \mathcal{Q}$ arise. We specify the following convenient exponential family distributions for each approximating density function:

$$\begin{gathered} \mathfrak{q}(\boldsymbol{\beta}) \text{ is a } \mathrm{N}\big(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})},\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta})}\big) \text{ density function}, \\ \mathfrak{q}(\sigma^2) \text{ is an Inverse-Gamma}\big(\xi_{\mathfrak{q}(\sigma^2)},\lambda_{\mathfrak{q}(\sigma^2)}\big) \text{ density function}, \\ \mathfrak{q}(a) \text{ is an Inverse-Gamma}\big(\xi_{\mathfrak{q}(a)},\lambda_{\mathfrak{q}(a)}\big) \text{ density function}. \end{gathered} \tag{2.20}$$

It is extremely important not to specify further inner partitions for $\boldsymbol{\beta}$, and therefore to jointly approximate the full parameter $\boldsymbol{\beta}$ in order to account for the dependence relationships among them with the covariance matrix $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta})}$. Otherwise, if further factorizations are considered, e.g. $\{\boldsymbol{\beta}\} \to \{\{\beta_0\},\ldots,\{\beta_{d-1}\}\}$, then $\mathfrak{q}(\boldsymbol{\beta}) = \prod_{h=0}^{d-1}\mathfrak{q}(\beta_h)$ and this automatically implies the conditional independence among all the regression coefficients. This leads to inaccurate estimation of the variability for

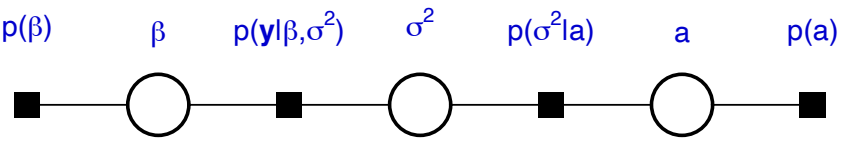

FIGURE 2.2: *Factor graph representation of model* (2.19) *following parameter partition* $\{\{\boldsymbol{\beta}\},\{\sigma^2\},\{a\}\}$ *and joint density factorization* $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\mathfrak{p}(\boldsymbol{\beta})\mathfrak{p}(\sigma^2|a)\mathfrak{p}(a)$.

the marginal posterior distributions on each regression coefficient, and hence to possibly misleading inferential conclusions.

Figure 2.2 shows the factor graph representation of model (2.19) with the pre-specified parameter partition. Notice its structure exactly corresponds to the factor graph displayed in Figure 2.1 for the univariate Normal random sample model, with μ replaced by $\boldsymbol{\beta}$. The messages involving factors $\mathfrak{p}(\sigma^2|a)$ and $\mathfrak{p}(a)$ remain identical to those described for that model and do not need to be derived again. In fact, the same hierarchical prior construction is placed on σ and the same exponential family distributions are chosen for $\mathfrak{q}(\sigma^2)$ and $\mathfrak{q}(a)$.

Yet, a new degree of complexity is introduced here, since Kullback-Leibler projections onto the multivariate Normal distribution family are needed for determining and updating both the $\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}(\boldsymbol{\beta})$ and $\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}(\boldsymbol{\beta})$ messages. In particular, this would require the computation of

$$\int_{\mathbb{R}^d} h(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta}, \qquad \int_{\mathbb{R}^d} \boldsymbol{\beta}\, h(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta} \quad \text{and} \quad \int_{\mathbb{R}^d} \mathrm{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T)\, h(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta}$$

for some $h(\boldsymbol{\beta})$, arising numerical intractability even for $d > 3$. Kim and Wand (2018) performed EP approximations treating our same model specification, although considering a different factor graph structure with n factors in place of $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)$, one for each likelihood term $\mathfrak{p}(y_i|\boldsymbol{\beta},\sigma^2)$. They circumvented this computational problem by introducing n auxiliary variables $\alpha_1,\ldots,\alpha_n$ (and n additional factors in the underlying factor graph as a direct consequence) each having a degenerate Dirac density function of the form $\delta(\alpha_i - \boldsymbol{x}_i^T\boldsymbol{\beta})$. Such a formulation allowed them to directly update all the natural parameter vectors involving the factor $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)$, but suffers two major limitations. Firstly, Power-EP approximations can not be formulated generalizing their results, as this would require the mathematical definition of $\delta(\cdot)^\alpha$ for any $\alpha \in (0,1]$. Secondly, it requires the update of $4n$ factor to stochastic node messages ($2n$ for all the $\mathfrak{p}(y_i|\boldsymbol{\beta},\sigma^2)$ factors, and $2n$ for all the $\delta(\alpha_i - \boldsymbol{x}_i^T\boldsymbol{\beta})$ factors) on each iteration, slowing down the overall algorithm runtime when large sample sizes are considered.

We propose a different strategy that overcomes these two limitations, making explicit reference to the factor graph structure displayed in Figure 2.2 and only employing standard univariate numerical techniques. Following the same step proposed for the univariate Normal random sample model, a proper Power-EP message-passing algorithm iterates over natural parameter vectors updates (2.9)–(2.10) until convergence, returning the optimal natural parameter vectors $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}$, $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)}$ and $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)}$. The associated optimal $\mathfrak{q}$-densities were specified as:

$$\begin{aligned}
&\mathfrak{q}^*_\alpha(\boldsymbol{\beta}) \text{ is the } \mathrm{N}\big(-\{\mathrm{vech}^{-1}([\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}]_2)\}^{-1}[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}]_1/2, -\{\mathrm{vech}^{-1}([\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}]_2)\}^{-1}/2\big) \\
&\qquad\qquad \text{density function ,} \\
&\mathfrak{q}^*_\alpha(\sigma^2) \text{ is the Inverse-Gamma}\big(-[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)}]_1 - 1, -[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)}]_2\big) \text{ density function ,} \\
&\mathfrak{q}^*_\alpha(a) \text{ is the Inverse-Gamma}\big(-[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)}]_1 - 1, -[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)}]_2\big) \text{ density function .}
\end{aligned} \tag{2.21}$$

Unlike for Power-EP approximations on the univariate Normal random sample model, here $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ result from a Kullback-Leibler projection onto the multivariate Normal exponential family of distributions. The following result extends Result 2.1 to give an operational strategy for determining such natural parameter vector:

Result 2.3. *Let $\boldsymbol{X}$ be a non-degenerate d-dimensional random vector for which $E(\boldsymbol{X}\boldsymbol{X}^T)$ exists and with density function $\mathfrak{p}(\boldsymbol{x})$. The Kullback-Leibler projection of $\mathfrak{p}(\boldsymbol{x})$ onto the d-dimensional multivariate Normal family,*

$$\mathfrak{q}_{KL}(\boldsymbol{x};\boldsymbol{\mu}_{KL},\boldsymbol{\Sigma}_{KL}) = proj_{MVN_d}[\mathfrak{p}(\boldsymbol{x})] = \underset{\mathfrak{q}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})\in N(\boldsymbol{\mu},\boldsymbol{\Sigma})}{\arg\min}\ \mathcal{KL}\{\mathfrak{p}(\boldsymbol{x})\|\mathfrak{q}(\boldsymbol{x};\mu,\boldsymbol{\Sigma})\}.$$

is the $N(\boldsymbol{\mu}_{KL},\boldsymbol{\Sigma}_{KL})$ density function with mean vector and covariance matrix

$$\boldsymbol{\mu}_{KL} = E(\boldsymbol{X}) \quad \textit{and} \quad \boldsymbol{\Sigma}_{KL} = E(\boldsymbol{X}\boldsymbol{X}^T) - \boldsymbol{\mu}_{KL}\boldsymbol{\mu}^T_{KL}$$

respectively.

Computation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a}$ requires a Kullback-Leibler projection onto the Inverse-Gamma (Inverse-χ^2) exponential family of distributions and an operational strategy again follows from Result 2.2.

Algorithm 2.3 gives a synthetic and general scheme for performing Power-EP approximations on model (2.19) with approximating density functions restricted to the parametric family described in (2.20). Without loss of generality, we initialize

Algorithm 2.3 *Power Expectation Propagation message-passing algorithm on factor graph displayed in Figure 2.2 for determining the natural parameter vectors of the optimal density functions* (2.21) *for approximate Bayesian inference on model* (2.19).

Data Inputs: $\boldsymbol{y}$ $(n \times 1)$, $\boldsymbol{X}$ $(n \times d)$.

Create summary data vector $\boldsymbol{D} = (n, \|\boldsymbol{y}\|^2, (\boldsymbol{X}^T\boldsymbol{y})^T, -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X})^T)^T$.

Power-EP α choice: $\alpha \in (0, 1]$.

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}} \in \mathbb{R}^d$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} \in \mathbb{S}_+^d$ and $s > 0$.

Initialize: $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} \leftarrow \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} \leftarrow \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix}$,

$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} \leftarrow \begin{bmatrix} \boldsymbol{0} \\ -\frac{1}{2}\text{vech}(\boldsymbol{I}) \end{bmatrix}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} \leftarrow \begin{bmatrix} -2 \\ -1 \end{bmatrix}$,

$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \leftarrow \begin{bmatrix} -2 \\ -1 \end{bmatrix}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \leftarrow \begin{bmatrix} -2 \\ -1 \end{bmatrix}$.

$\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)} \leftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$; $\boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(\sigma^2|a)} \leftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}$.

Cycle until convergence:

Updates for the $\mathfrak{p}(\boldsymbol{\beta})$ factor: fixed.

Updates for the $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)$ factor:

$$\boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} \longleftarrow G^{\text{MVN}}_{\text{lm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}'}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}; \alpha\boldsymbol{D}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} \longleftarrow G^{\text{IG4}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}}; \alpha\boldsymbol{D}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}.$$

Updates for the $\mathfrak{p}(\sigma^2|a)$ factor:

$$\boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\sigma^2|a)} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \longleftarrow G^{\text{IG2}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}, \begin{bmatrix} [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}]_1 + 2(1-\alpha) \\ [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}]_2/\alpha \end{bmatrix}; 3\alpha\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2};$$

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \longleftarrow G^{\text{IG2}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a'}, \begin{bmatrix} [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}]_1 + 2(1-\alpha) \\ [\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}]_2/\alpha \end{bmatrix}; \alpha\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}.$$

Updates for the $\mathfrak{p}(a)$ factor: fixed.

Outputs: $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}'}$, $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\sigma^2)} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$,

$\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(a)} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$.

$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ to be the natural parameter vector of a d-dimensional $\text{N}(\boldsymbol{0}, \boldsymbol{I})$ distribution and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$ to be the natural parameter vector of an Inverse-Gamma$(1, 1)$

distribution, while $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ follow the same initialization of Algorithm 2.2. Details on the $G^{\text{MVN}}_{\text{lm}}$, G^{IG1}, G^{IG2} and G^{IG4} wrappers and derivations of the associated Kullback-Leibler projected natural parameter vectors are given in Appendices B.1 and B.2, respectively.

Comments similar to those given for Power-EP on the univariate Normal random sample model apply here. In particular, notice how all the data information required for performing the updates can be synthesized into the $\boldsymbol{D}$ vector, and how α acts as a weight for its contribution to the Power-EP updates involving the $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)$ factor. Moreover, the beauty of the message-passing algorithmic infrastructure can be captured in this example: Algorithm 2.3 is just a modified version of Algorithm 2.2 in which we account for a different form of the likelihood-associated factor, and hence only its associated message updates need to be derived from scratch.

The major drawback of Algorithm 2.3 is that it requires the numerical computation of $d + d(d+1)/2$ univariate integrals for performing the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ update at each iteration. This task grows as $\mathcal{O}(d^2)$ and, as such, can sensibly slow down the overall runtime of the estimating algorithm when the number of regressors considered is far from being small. The choice of which algorithm to employ depends on the data dimensions and the desired approximation type. Algorithm 1 of Kim and Wand (2018) can only perform EP approximation and is preferable when the sample size is small or moderate but allows for large covariate dimensions. On the other hand, Algorithm 2.3 can perform Power-EP approximations for any $\alpha \in (0,1]$ and scales efficiently for large sample sizes, but does not scale well for large d.

2.5 Power-EP for Some Notable GLMs

In many applicative contexts, the response vector $\boldsymbol{y}$ cannot be treated as being generated from a multivariate Normal distribution, e.g., when it refers to a binary or count variable of interest. Therefore, model (2.19) is no more applicable for investigating how the associated response variable links to a set of possibly related regressors. The most employed alternative proposed in statistical literature refers to the generalized linear models (GLMs) class. Hereafter, we restrict the focus to Bayesian GLMs having general specifications:

$$\begin{gathered} Y_i|\boldsymbol{\beta} \overset{\text{ind}}{\sim} \mathfrak{p}(y_i|\boldsymbol{\beta}) = \exp\left\{y_i A'^{-1}(F(\boldsymbol{x}_i^T\boldsymbol{\beta})) - A(A'^{-1}(F(\boldsymbol{x}_i^T\boldsymbol{\beta})))\right\} h(y_i), \\ \boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}) \end{gathered} \tag{2.22}$$

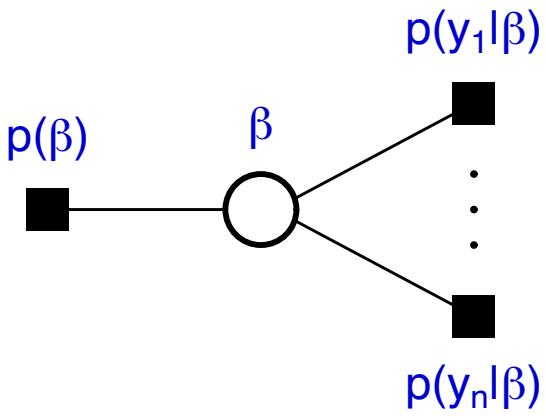

FIGURE 2.3: *Factor graph representation of model* (2.22) *with joint density factorization* $\mathfrak{p}(\boldsymbol{\beta}) \prod_{i=1}^{n} \mathfrak{p}(y_i|\boldsymbol{\beta})$. *The vertical dots indicates that n different factors* $\mathfrak{p}(y_i|\boldsymbol{\beta})$ *are considered, all sharing an edge with stochastic node* $\boldsymbol{\beta}$.

for each $1 \leq i \leq n$. Here $A(\eta)$ is the log-partition function of a suitably chosen univariate distribution belonging to the exponential family (see Appendix A.2), $F(\eta)$ is the inverse of the so-called *link function* and $A'^{-1}(\eta)$ represents the inverse of $A'(\eta) = \mathrm{d}A(\eta)/\mathrm{d}\eta$. This formulation explicitly shows how the d-dimensional parameter vector $\boldsymbol{\beta}$ interacts with the unit-associated regressor vector $\boldsymbol{x}_i$ in determining the likelihood expression for y_i, and reduces into a simpler and more intuitive form if $F(\eta) = A'(\eta)$, that is, if the *canonical* link function is adopted. A prior for $\boldsymbol{\beta}$ identical to that for the linear regression model specification (2.19) is chosen.

Specification (2.22) is a particular case of the wider *exponential dispersion family* from which GLMs arise, which consider an additional dispersion parameter. We show that restricting our work to this subfamily still accounts for the most common GLMs. Regardless of the distribution chosen for the response variable, the only parameter involved here is $\boldsymbol{\beta}$ and so the associated posterior probability density function is simply $\mathfrak{p}(\boldsymbol{\beta}|\boldsymbol{y}) \propto \mathfrak{p}(\boldsymbol{\beta}) \prod_{i=1}^{n} \mathfrak{p}(y_i|\boldsymbol{\beta})$.

Power-EP finds the optimal posterior approximating density $\mathfrak{q}^*_\alpha(\boldsymbol{\beta})$ that best solves (2.2) for a fixed $\alpha \in (0, 1]$: no additional factorizations for the generic $\mathfrak{q}(\boldsymbol{\beta}) \in \mathcal{Q}$ are needed, so as not to introduce misleading posterior conditional independence between the coefficients. Therefore, we fix the following parametric restriction for the generic $\mathfrak{q}$-density:

$$\mathfrak{q}(\boldsymbol{\beta}) \text{ is a } \mathrm{N}\big(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta})}\big) \text{ density function .} \tag{2.23}$$

Figure 2.3 shows the factor graph representation of model (2.22). Notice here we specifically treat each of the n likelihood factors separately, differently to the joint

likelihood factor displayed in Figures 2.1 and 2.2. Although this choice yields $2n$ natural parameter vectors to be updated at each iteration, we show hereafter that this allows the overall Power-EP approximating algorithm to be performed only employing univariate numerical integrals.

As well as for model (2.19), the Kullback-Leibler projections onto the multivariate Normal distribution family must be performed in order to update each of the $\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}(\boldsymbol{\beta})$ messages, requiring d-variate numerical integrals to be computed. A first attempt to solve this drawback was discussed in Section 4 of Kim and Wand (2018) employing the aforementioned derived variables trick, despite maintaining all the afore discussed associated problems. Moreover, if applied to get EP approximations over GLMs, it does not yield explicit natural parameter vector update expressions anymore.

We propose instead a new and totally different approach inspired by Hall *et al.* (2020) and associated Lemma 1 given in Section S.1 of their supplementary materials, which we rephrase here in a more convenient form:

Result 2.4. *For any function $g : \mathbb{R} \longrightarrow \mathbb{R}$, scalar $a \in \mathbb{R}$ and vector $\boldsymbol{b}$ of dimension $d \times 1$ such that the integrals exists, the following identity holds:*

$$\int_{\mathbb{R}^d} g(a + \boldsymbol{b}^T\boldsymbol{x})\, \phi(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \int_{-\infty}^{\infty} g(a + \|\boldsymbol{b}\|z)\, \phi(z)\, \mathrm{d}z,$$

$$\int_{\mathbb{R}^d} \boldsymbol{x}\, g(a + \boldsymbol{b}^T\boldsymbol{x})\, \phi(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|} \int_{-\infty}^{\infty} z\, g(a + \|\boldsymbol{b}\|z)\, \phi(z)\, \mathrm{d}z$$

and

$$\int_{\mathbb{R}^d} \boldsymbol{x}\boldsymbol{x}^T\, g(a + \boldsymbol{b}^T\boldsymbol{x})\, \phi(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \boldsymbol{I}_d \int_{-\infty}^{\infty} g(a + \|\boldsymbol{b}\|z)\, \phi(z)\, \mathrm{d}z + \frac{\boldsymbol{b}\boldsymbol{b}^T}{\|\boldsymbol{b}\|^2} \int_{-\infty}^{\infty} (z^2 - 1) g(a + \|\boldsymbol{b}\|z)\, \phi(z)\, \mathrm{d}z.$$

In short words, if $\boldsymbol{Z} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ then Result 2.4 simplifies the computation of $E(g(a + \boldsymbol{b}^T\boldsymbol{Z}))$, $E(\boldsymbol{Z}g(a + \boldsymbol{b}^T\boldsymbol{Z}))$ and $E(\boldsymbol{Z}\boldsymbol{Z}^T g(a + \boldsymbol{b}^T\boldsymbol{Z}))$ boiling down d-variate integrals into linear transformations of univariate integrals.

We focus our work on three of the most employed GLMs, namely the *probit* regression, the *logistic* regression, and the *Poisson* regression models. We present a Power-EP algorithm that approximates their respective posterior distributions for any $\alpha \in (0, 1]$. We group all the three associated resulting algorithms into Algorithm 2.4, because they share the same structure and only differ in the input arguments

Algorithm 2.4 *Power Expectation Propagation message-passing algorithm on factor graph displayed in Figure 2.3 for determining the natural parameter vectors of the optimal density functions* (2.24) *for approximate Bayesian inference on model* (2.22).

Data Inputs: $\boldsymbol{y}$ $(n \times 1)$, $\boldsymbol{X}$ $(n \times d)$.

Power-EP α choice: $\alpha \in (0, 1]$.

GLM regression type choice: Probit, Logistic or Poisson.

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}} \in \mathbb{R}^d$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} \in \mathbb{S}_+^d$.

Initialize: $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} \leftarrow \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\mathrm{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \leftarrow \begin{bmatrix} \boldsymbol{0} \\ -\frac{1}{2}\mathrm{vech}(\boldsymbol{I}) \end{bmatrix}$ for $1 \le i \le n$.

Cycle until convergence:

Update for the $\mathfrak{p}(\boldsymbol{\beta})$ factor:

$$\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{\beta})} \longleftarrow \sum_{i=1}^{n} \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$$

Updates for the $\mathfrak{p}(y_i|\boldsymbol{\beta})$ factor, for $1 \le i \le n$:

$$\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(y_i|\boldsymbol{\beta})} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} + \boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{\beta})} - \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$$

If Probit:

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow G^{\mathrm{MVN}}_{\mathrm{glm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}; 0, (2y_i - 1)\boldsymbol{x}_i, 0, -\alpha\log(\Phi(x))\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$$

If Logistic:

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow G^{\mathrm{MVN}}_{\mathrm{glm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}; 0, (2y_i - 1)\boldsymbol{x}_i, \alpha, \alpha\log(1+e^x)\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$$

If Poisson:

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow G^{\mathrm{MVN}}_{\mathrm{glm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}; 0, \boldsymbol{x}_i, \alpha y_i, \alpha e^x\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$$

Output: $\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})} = \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} + \sum_{i=1}^{n} \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$.

of the general $G^{\mathrm{MVN}}_{\mathrm{glm}}$ wrapper defined in Appendix B.1. Without loss of generality, all the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ vectors were initialized to be the natural parameter vectors of a $\mathrm{N}(\boldsymbol{0}, \boldsymbol{I})$ distribution, for $1 \le i \le n$, and once convergence is reached the optimal approximating $\mathfrak{q}$-density reads:

$$\mathfrak{q}^*_\alpha(\boldsymbol{\beta}) \text{ is the } \mathrm{N}\big(-\{\mathrm{vech}^{-1}([\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}]_2)\}^{-1}[\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}]_1/2, -\{\mathrm{vech}^{-1}([\boldsymbol{\eta}_{\mathfrak{q}^*_\alpha(\boldsymbol{\beta})}]_2)\}^{-1}/2\big) \text{ density function .} \tag{2.24}$$

Derivations of the required Kullback-Leibler projected natural parameter vectors are given in Appendix B.2. The major limitation of Algorithm 2.4 resides in the fact that each iteration cycles over n factors to be updated iteratively, one for each

observation. Nonetheless, it works for any $\alpha \in (0, 1]$ and is very efficient even for moderate to large covariate dimensions. The reason is due to the use of the $G_{\text{glm}}^{\text{MVN}}$ wrapper, which only performs univariate numerical integrals exploiting the massive contribution of Result 2.4. Appendix B.2 provides further explanations on how Result 2.4 can be used to boil down the dimension of the numerical integrals required for performing the natural parameter vector updates. Comparing our work with that presented in Section 4 of Kim and Wand (2018), we obtain their identical approximation strategy, albeit it does not require the introduction of n additional derived variables and associated $2n$ additional natural parameter vector updates to be performed at each iteration. Moreover, their approach is only valid for EP approximations, while ours treats it as a particular case that can be recovered by selecting $\alpha = 1$ before starting Algorithm 2.4.

We now give some insights into the three different GLM models considered, showing how they belong to the general model specification (2.22). We also add closed-form update expressions not requiring any numerical integration methods to be performed, arising for EP approximations over the probit and logistic regression models.

2.5.1 Probit Regression

Probit regression models arise if $y_i \in \{0, 1\}$ and a Bernoulli distribution for the i-th response variable is assumed, having $A(\eta) = \log(1 + e^{\eta})$ and $h(y) = 1$, with $F(\eta) = \Phi(\eta)$ chosen as link function:

$$Y_i|\boldsymbol{\beta} \overset{\text{ind}}{\sim} \text{Bernoulli}(\Phi(\boldsymbol{x}_i^T\boldsymbol{\beta})).$$

With such a specification, (2.22) can be rewritten in a more compact form as $\mathfrak{p}(y_i|\boldsymbol{\beta}) = \Phi((2y_i - 1)\boldsymbol{x}_i^T\boldsymbol{\beta})$.

2.5.1.1 Closed Form EP Update

Power-EP approximate inference for the Bayesian probit regression model admits a closed-form update expression if the EP approximation is required. Hall *et al.* (2020) gave the generic idea, albeit from a frequentist point of view and on a probit mixed-effects regression model. In their Section 3.1 and associated results, they were able to derive an exact and explicit expression for the Kullback-Leibler projection of the generic probit likelihood term onto the multivariate Normal distribution family, employing their Result 1 together with theoretical properties of the inverse

link function $\Phi(\eta)$. Such a derivation does not require any numerical integration methods and remarkably improves the overall computational time for the EP algorithm. Making explicit usage of the K_{probit} wrapper introduced in their Definition 1, we can substitute the natural parameter vector update expression of Algorithm 2.4 with

$$\boldsymbol{\eta}^{(1)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow K_{\text{probit}}\left(\boldsymbol{\eta}^{(1)}_{\boldsymbol{\beta}\to\mathfrak{p}(y_i|\boldsymbol{\beta})}; 0, (2y_i-1)\boldsymbol{x}_i\right) - \boldsymbol{\eta}^{(1)}_{\boldsymbol{\beta}\to\mathfrak{p}(y_i|\boldsymbol{\beta})},$$

which only requires matrix multiplications and inversions to be performed with standard algebraic routines. Derivation of this novel expression for $\boldsymbol{\eta}^{(1)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ follows immediately by fixing $\alpha = 1$ and employing Theorem 1 of Hall *et al.* (2020) into derivations for the probit Power-EP algorithm presented in Appendix B.2.

2.5.2 Logistic Regression

Logistic regression models arise if $y_i \in \{0, 1\}$ and a Bernoulli distribution for the i-th response variable is assumed, having $A(\eta) = \log(1 + e^{\eta})$ and $h(y) = 1$, with $F(\eta) = \text{expit}(\eta)$ being the canonical link function:

$$Y_i|\boldsymbol{\beta} \overset{\text{ind}}{\sim} \text{Bernoulli}(\text{expit}(\boldsymbol{x}_i^T\boldsymbol{\beta})).$$

With such a specification, (2.22) can be rewritten in a more compact form as $\mathfrak{p}(y_i|\boldsymbol{\beta}) = \text{expit}((2y_i - 1)\boldsymbol{x}_i^T\boldsymbol{\beta})$.

2.5.2.1 Closed Form EP Update

Power-EP approximate inference for the Bayesian logistic regression model admits a closed-form update expression if the EP approximation is required. Monahan and Stefanski (1989) introduced a Normal scale mixture approximation of the $\text{expit}(\eta)$ function for which, given $k \in \mathbb{N}$,

$$\sup_{x\in\mathbb{R}}\left|\text{expit}(x) - \sum_{i=1}^{k} p_{k,i}\Phi(s_{k,i}\,x)\right| = \Delta_k$$

for appropriately chosen constants $p_{k,i}, s_{k,i}$ and resulting finite bound Δ_k. They tabulated explicit values up to $k = 8$, for which $\Delta_8 = 2.1 \times 10^{-9}$ and related constants $p_{k,i}, s_{k,i}$ are reported in Table 2.1. Such a result allows to rephrase the logistic natural parameter vector update in terms of a finite mixture of Normal cumulative density functions, yielding an explicit natural parameter update expression that does not need any univariate integration method to be performed.

i	$p_{8,i}$	$s_{8,i}$
1	0.00324 63432 72134	1.36534 08062 96348
2	0.05151 74770 33972	1.05952 39710 16916
3	0.19507 79126 73858	0.83079 13137 65644
4	0.31556 98236 32818	0.65073 21666 39391
5	0.27414 95761 58423	0.50813 54253 66489
6	0.13107 68806 95470	0.39631 33451 66341
7	0.02791 24187 27972	0.30890 42522 67995
8	0.00144 95678 05354	0.23821 26164 09306

TABLE 2.1: *Tabulated values for the $p_{k,i}$ and s_{k_i} costants given in Monahan and Stefanski (1989), corresponding to the $k = 8$ Normal scale mixture approximation for expit(x).*

For $\boldsymbol{a}_1$ $(d \times 1)$, $\boldsymbol{a}_2$ $(d(d+1)/2 \times 1)$ such that $\text{vech}^{-1}(\boldsymbol{a}_2)$ is symmetric and positive definite, $c_0 \in \mathbb{R}$ and $\boldsymbol{c}_1$ $(d \times 1)$, define the K_{logistic} wrapper:

$$K_{\text{logistic}}\left(\left[\begin{array}{c}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{array}\right]; c_0, \boldsymbol{c}_1\right) \equiv \left[\begin{array}{c}\boldsymbol{R}_5^T(\boldsymbol{a}_1 + r_3\boldsymbol{c}_1)\\ \text{vech}(\boldsymbol{R}_5^T\boldsymbol{A}_2)\end{array}\right]$$

with

$$\boldsymbol{A}_2 \equiv \text{vech}^{-1}(\boldsymbol{a}_2), \quad r_{i1} \equiv \sqrt{2\left(2 - s_{8,i}^2(\boldsymbol{c}_1^T\boldsymbol{A}_2^{-1}\boldsymbol{c}_1)\right)},$$

$$r_{i2} \equiv s_{8,i}(2c_0 - \boldsymbol{c}_1^T\boldsymbol{A}_2^{-1}\boldsymbol{a}_1)/r_{i1}, r_3 \equiv \frac{2\sum_{i=1}^8(p_{8,i}s_{8,i}\phi(r_{i2})/r_{i1})}{\sum_{i=1}^8(p_{8,i}\Phi(r_{i2}))},$$

$$r_4 \equiv \frac{r_3^2}{2} + \frac{2\sum_{i=1}^8(p_{8,i}s_{8,i}^2 r_{i2}\phi(r_{i2})/r_{i1}^2)}{\sum_{i=1}^8(p_{8,i}\Phi(r_{i2}))}, \quad \boldsymbol{R}_5 \equiv (\boldsymbol{A}_2 + r_4\boldsymbol{c}_1\boldsymbol{c}_1^T)^{-1}\boldsymbol{A}_2$$

for $1 \leq i \leq 8$, where the $p_{8,i}$ and $s_{8,i}$ constants are defined in Table 2.1. Then we can substitute the natural parameter vector update expression of Algorithm 2.4 with

$$\boldsymbol{\eta}^{(1)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow K_{\text{logistic}}\left(\boldsymbol{\eta}^{(1)}_{\boldsymbol{\beta}\to\mathfrak{p}(y_i|\boldsymbol{\beta})}; 0, (2y_i - 1)\boldsymbol{x}_i\right) - \boldsymbol{\eta}^{(1)}_{\boldsymbol{\beta}\to\mathfrak{p}(y_i|\boldsymbol{\beta})},$$

which only requires matrix multiplications and inversions to be performed with standard algebraic routines.

This update expression is obtained employing an approximation of the expit(x) function and therefore it is not a purely exact result. Nonetheless, it can be performed without employing any numerical integration methods and therefore it remarkably improves the overall computational time for computing EP approximations. The expression for the K_{logistic} wrapper and update of $\boldsymbol{\eta}^{(1)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ arise naturally after substituting $\sum_{i=1}^k p_{k,i}\Phi(s_{k,i}\, x)$ to the expit(x) function into the derivation

for the logistic Power-EP algorithm, after fixing $\alpha = 1$ and rearranging the involved terms with simple algebraic operations.

2.5.3 Poisson Regression

Poisson regression models arise if $y_i \in \mathbb{N}$ and a Poisson distribution for the i-th response variable is assumed, having $A(\eta) = e^{\eta}$ and $h(y) = 1/y!$, with $F(\eta) = \exp(\eta)$ chosen as link function:

$$Y_i|\boldsymbol{\beta} \overset{\text{ind}}{\sim} \text{Poisson}(\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})).$$

With such a specification, (2.22) can be rewritten in a more compact form as $\mathfrak{p}(y_i|\boldsymbol{\beta}) = \exp\{y_i\boldsymbol{x}_i^T\boldsymbol{\beta} - \exp(\boldsymbol{x}_i^T\boldsymbol{\beta})\}\backslash y_i!$. Differently from the probit and logistic regressions, no explicit Power-EP natural parameter vector update expressions arise in the particular $\alpha = 1$ case.

2.6 Numerical Investigations on Simulated Data

We now report the results of a simulation study conducted for investigating how some of the Power-EP algorithms presented so far perform and how accurately they approximate the true posterior density functions, for different $\alpha \in (0, 1]$ values. All the algorithms have been implemented in pure R language. Moreover, 10^5 samples from the *true* posterior density functions were obtained via MCMC with the `rstan` package (see Appendix A.3), after a burnin of size 10^3.

The simulation study has been conducted as follows: we chose five different sample size scenarios $n \in \{25, 50, 100, 500, 1000\}$ and, for each of them, we generated 100 data replicates accordingly to the statistical model specification considered, for some fixed true parameter values. Then, for each data replication we ran the Power-EP algorithm for four different $\alpha \in \{0.25, 0.5, 0.75, 1\}$ values selected in the unit interval. The convergence was assessed whenever

$$\max_{1 \leq i \leq M} \left| \frac{\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\theta}_i,\text{curr}}}{\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\theta}_i,\text{prev}}} - 1 \right| < 10^{-4}$$

with $\boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\theta}_i} \equiv \sum_{j:i\in\text{neigh}(j)} \boldsymbol{\eta}^{(\alpha)}_{f_j \to \boldsymbol{\theta}_i'}$ meaning that the one further iteration of the message-passing algorithm does not bring a significant relative change in any of the resulting natural parameter vectors for $\mathfrak{q}^*_\alpha(\boldsymbol{\theta}_i)$, $1 \leq i \leq M$. We denote with the "curr" and

"prev" subscripts the natural parameter vector values corresponding to the current and previous iterations of the algorithm, respectively.

Moreover, we obtained MFVB approximations to compare with Power-EP approximations directly. Just notice for the GLM models considered in this chapter many competing alternative MFVB algorithms have been developed during the years, and we opted to use the most immediate ones to implement. We obtained five different sets of optimal approximating densities' parameters for each model parameter of interest: four for the different Power-EP approximations and one for the MFVB approximation. For each of them, we evaluated the accuracy index (1.25) to measure the goodness of the obtained approximation.

Some model-related choices have been made:

- Normal random sample model: we fixed $\mu_\mu = 0$, $\sigma_\mu = 10^5$ and $s = 10^5$ as prior hyperparameters, embedding uninformative prior specifications for all the model parameters. Each data sample corresponds to a vector $\boldsymbol{y}$ of n independently-generated draws from a univariate Normal distribution with true parameter values $\mu = 0$ and $\sigma^2 = 1$. The Power-EP optimal approximating densities have been found employing Algorithm 2.2. The MFVB optimal approximating densities have been found employing Algorithm 1 of Luts *et al.* (2014), with $\boldsymbol{X} = \mathbf{1}$ and $\boldsymbol{\beta} = \mu\mathbf{1}$.

- Normal linear regression model: we fixed $d = 5$ as the number of regressors considered and $\mu_{\boldsymbol{\beta}} = \mathbf{0}$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = 10^5\boldsymbol{I}$, $s = 10^5$ as prior hyperparameters, embedding uninformative prior specifications for all the model parameters. Each data sample corresponds to the same matrix $\boldsymbol{X}$ of dimension $n \times d$ with the first column corresponding to $\mathbf{1}$ and the remaining sub-block having each row generated from a multivariate Normal distribution having zero mean vector and covariance matrix generated from a Wishart$(d-1, \boldsymbol{I}_{d-1})$ distribution, and a response data vector $\boldsymbol{y}$ of length n generated from a multivariate Normal distribution having mean vector $\boldsymbol{X\beta}$ and covariance matrix $\sigma^2\boldsymbol{I}$, with true parameter values $\boldsymbol{\beta} = (-2.3, 1.4, -0.9, 2.1, 0.2)^T$ and $\sigma^2 = 1$. This choice mimics as much as possible a real data scenario in which regressors share some correlation structure and different variabilities. The Power-EP optimal approximating densities have been found employing Algorithm 2.3. Without loss of generality, all the columns of $\boldsymbol{X}$ have been normalized before running the algorithm. The MFVB optimal approximating densities have been found employing Algorithm 1 of Luts *et al.* (2014).

- Generalized linear models: we fixed $d = 5$ as the number of regressors considered and $\mu_\beta = \mathbf{0}$, $\Sigma_\beta = 10^5 \boldsymbol{I}$ as prior hyperparameters, embedding uninformative prior specifications for $\boldsymbol{\beta}$. Each data sample corresponds to the same matrix $\boldsymbol{X}$ of dimension $n \times d$ generated with the same identical strategy followed for the Normal linear model, and a response data vector $\boldsymbol{y}$ of length n generated from a model-specific distribution with linear predictor $\boldsymbol{X\beta}$, having true parameter value $\boldsymbol{\beta} = (-2.3, 1.4, -0.9, 2.1, 0.2)^T$. Differently from the Normal linear model specification, all the columns of $\boldsymbol{X}$ have been normalized before the generation of $\boldsymbol{y}$ as to avoid numerical underflows that could emerge from unreal generated data scenarios (e.g. binary data having response variable all equal to 0 or 1, or count data having response variable taking values particularly far from 0). The Power-EP optimal approximating densities have been found employing Algorithm 2.4. Moreover:

  - Probit regression: $\boldsymbol{y}$ was generated from the Bernoulli$(\Phi(\boldsymbol{X\beta}))$ distribution, and when $\alpha = 1$ is selected, the Power-EP algorithm switched to the closed-form update expression described before. The optimal MFVB approximations have been obtained employing Algorithm 4 of Ormerod and Wand (2010). It implements an approach developed by Girolami and Rogers (2006) and Consonni and Marin (2008) which is based on the auxiliary variable representation of Albert and Chib (1993). Nonetheless, such an approximation refers to a more general posterior density function including the vector of augmented variables, and therefore it is not fully comparable with the Power-EP approximations for $\mathfrak{p}(\boldsymbol{\beta}|\boldsymbol{y})$.
  - Logistic regression: $\boldsymbol{y}$ was generated from the Bernoulli$(\text{expit}(\boldsymbol{X\beta}))$ distribution, and when $\alpha = 1$ is selected, the Power-EP algorithm switched to the closed-form update expression described before. The MFVB optimal approximating densities have been obtained employing Algorithm 2 of Nolan and Wand (2017), which unifies results from Knowles and Minka (2011) and Wand (2014) for providing better approximation accuracy than the standard Jaakkola and Jordan (2000) approach.
  - Poisson regression: $\boldsymbol{y}$ was generated from the Poisson$(\exp(\boldsymbol{X\beta}))$ distribution. The MFVB optimal approximating densities have been obtained employing Algorithm 1 of Luts and Wand (2015) under minor modifications for excluding the random effect structure from their model specification.

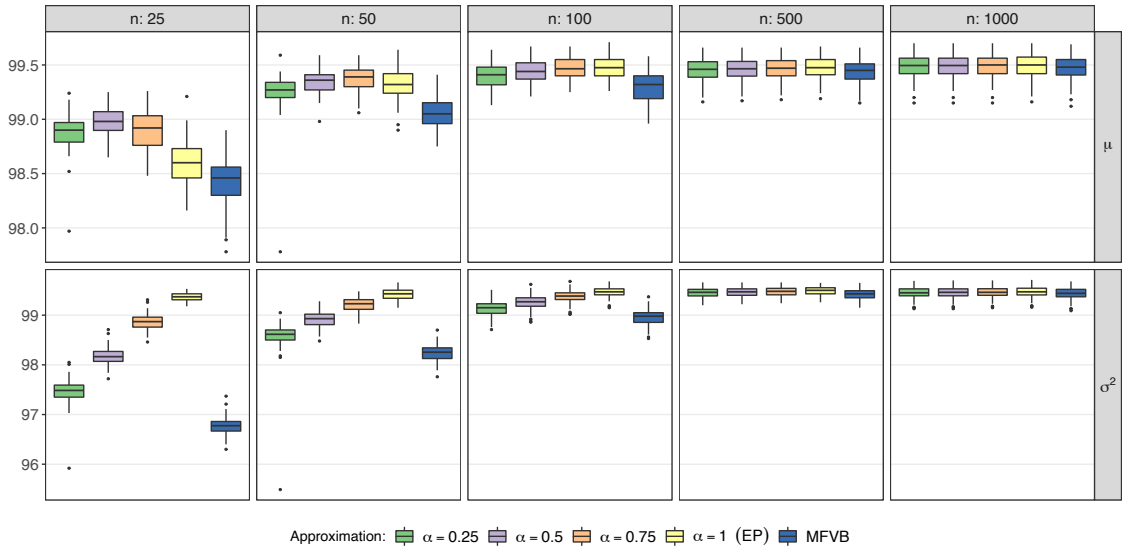

FIGURE 2.4: *Accuracy values for the optimal approximate densities of the univariate Normal random sample model, obtained from the simulation study. The first row corresponds to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\mu)$ *and* $\mathfrak{q}^*_{MFVB}(\mu)$ *to* $\mathfrak{p}(\mu|\boldsymbol{y})$*, while the second row corresponds to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\sigma^2)$ *and* $\mathfrak{q}^*_{MFVB}(\sigma^2)$ *to* $\mathfrak{p}(\sigma^2|\boldsymbol{y})$*. Different choices of* α *are indicated in the legend.*

A visual summarization of the results obtained for the statistical models considered in this work are reported in Figure 2.4 (Normal random sample model), Figure 2.5 (Normal linear regression model), Figure 2.6 (Probit regression model), Figure 2.7 (Logit regression model) and Figure 2.8 (Poisson regression model). All the figures are shown in the present and subsequent pages for clarity of exposition. For each Figure, a boxplot represents the distribution of the 100 accuracy scores obtained. Different filling colors correspond to the different variational approximation techniques employed, as described in the associated legend. A fixed y-scale is kept for each row of subfigures, aiding immediate comparison on how the accuracy scores increase as the sample size gets bigger and bigger.

All the approximating methods exhibit excellent accuracy performances for all the models considered. As expected, differences between the competing approximating methods vanish as the sample size increases. Figure 2.4 shows that, for small sample sizes, Power-EP with α values between 0 and 1 leads to slightly improved approximations for μ if compared to the standard EP and MFVB approximations. Nonetheless, for the same model, EP finds the best approximation for σ^2. A similar comment applies to the β coefficients of the linear regression model, see Figure 2.5. For all the GLMs considered, Power-EP approximations are almost always slightly more advantageous than the MFVB approximations, especially in the lower sample

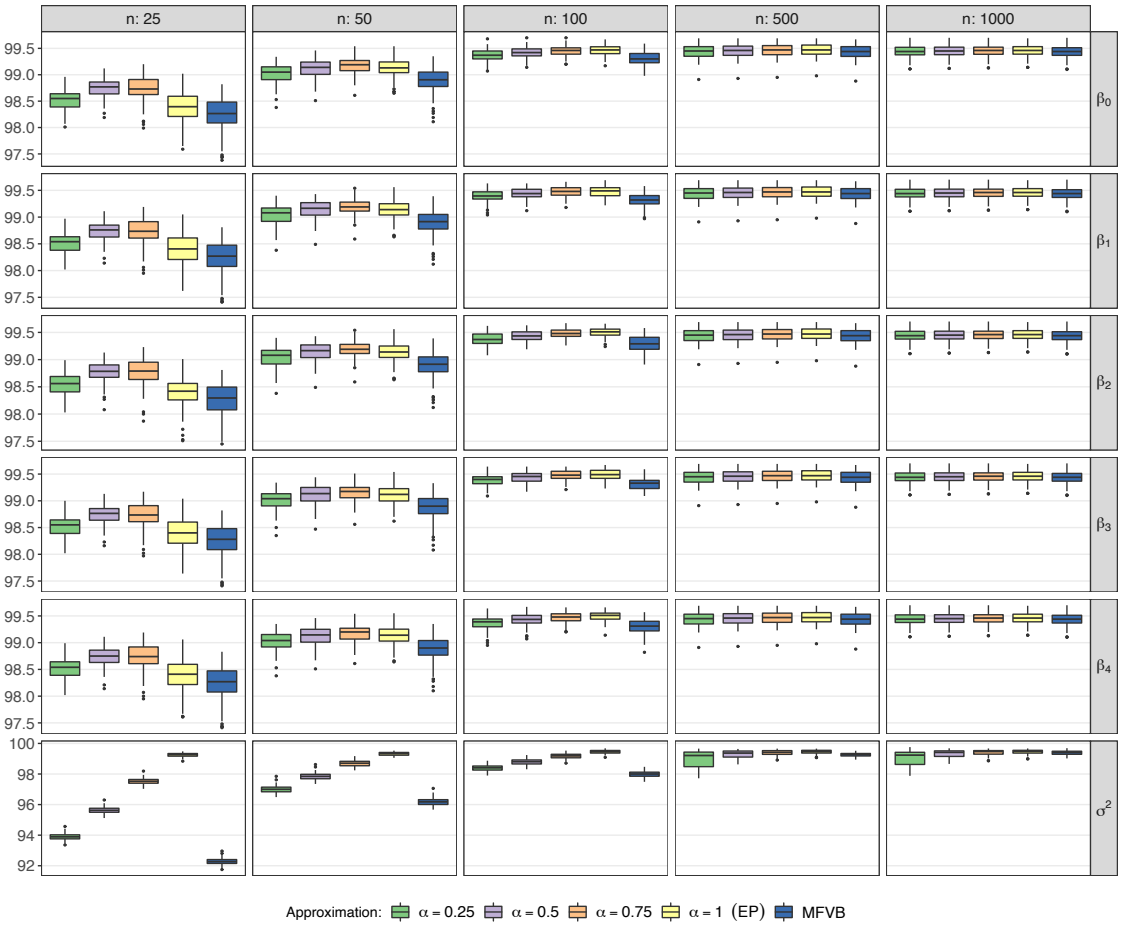

FIGURE 2.5: *Accuracy values for the optimal approximate densities of the Normal linear regression model, obtained from the simulation study. The first five rows correspond to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\beta_h)$ *and* $\mathfrak{q}^*_{MFVB}(\beta_h)$ *to* $\mathfrak{p}(\beta_h|\boldsymbol{y})$ *for* $0 \leq h \leq 4$*, while the last row corresponds to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\sigma^2)$ *and* $\mathfrak{q}^*_{MFVB}(\sigma^2)$ *to* $\mathfrak{p}(\sigma^2|\boldsymbol{y})$*. Different choices of* α *are indicated in the legend.*

scenarios where the MFVB algorithm may not have reached convergence successfully. Nonetheless, EP outperforms any other Power-EP approximation obtained by selecting $\alpha \in (0, 1)$.

In all the models considered, Power-EP approximations and their notable $\alpha = 1$ EP case show slightly better accuracy scores than the MFVB approximations. When considering low sample sizes, α values lower than one are sometimes especially useful. One possible motivation regards their ability of re-weighting the natural parameter vectors obtained from the Kullback-Leibler projection operator with a $(1 - \alpha) \times 100\%$ portion of its value obtained in the preceding iteration (just remember $\boldsymbol{\eta}^{(\alpha)}_{\text{factor}\to\text{node}} \longleftarrow G(\cdot) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\text{factor}\to\text{node}}$ for a proper $G(\cdot)$ wrapper function); if EP approximations are considered, this does not happen. Such projections are always based on univariate integrals computed with fast numerical routines, which

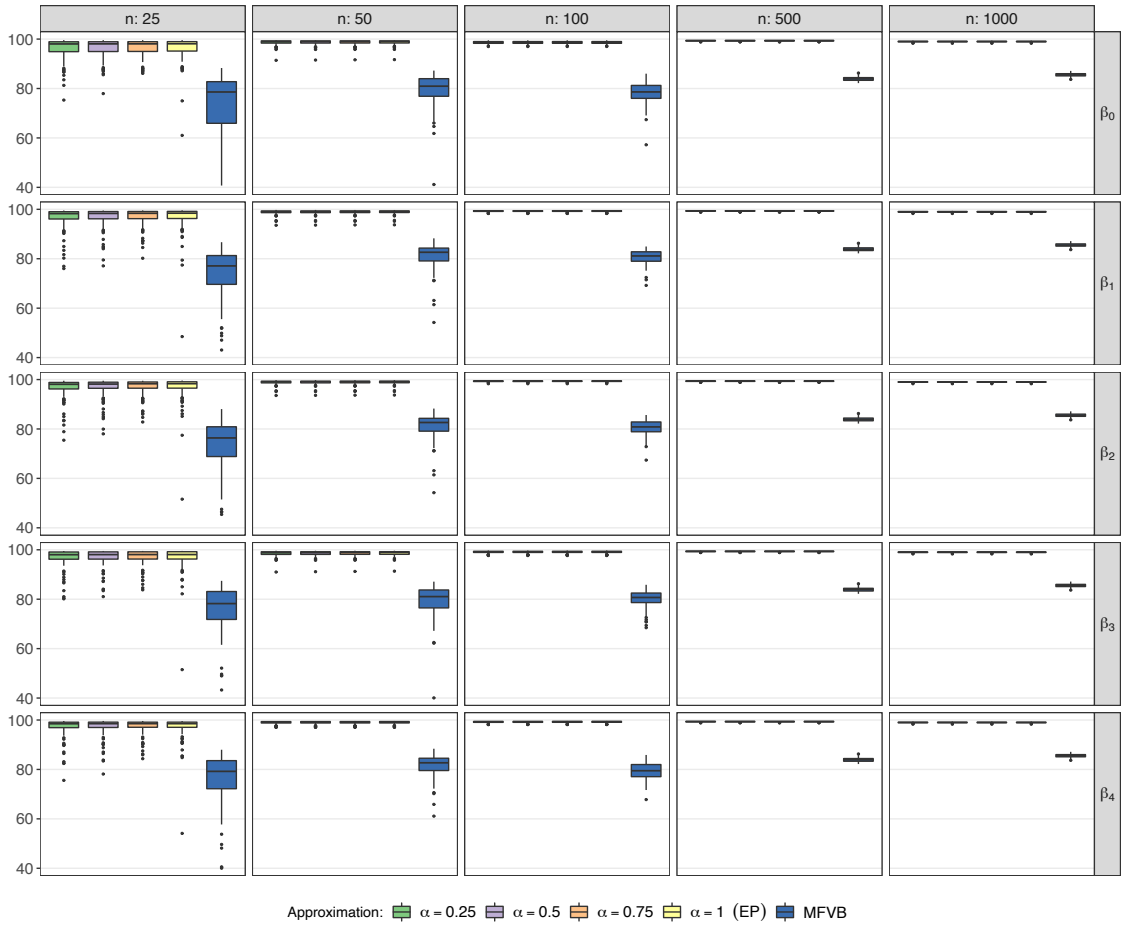

FIGURE 2.6: *Accuracy values for the optimal approximate densities of the probit regression model, obtained from the simulation study. Each row corresponds to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\beta_h)$ *and* $\mathfrak{q}^*_{MFVB}(\beta_h)$ *to* $\mathfrak{p}(\beta_h|\boldsymbol{y})$ *for* $0 \leq h \leq 4$*. Different choices of* α *are indicated in the legend.*

may lead to sensibly incorrect approximations for low sample sizes. Power-EP mitigates this unappealing possibility slowing down the EP algorithm from directing itself towards incorrect stationary solutions. Interestingly, Minka (2005) introduced the *damping* adjustment procedure recommending it when approximating algorithms do not converge properly, although not giving any theoretically-founded motivations. Power-EP updates' structures derived in this work mimic such adjustment, providing a newer motivation in terms of α-divergence minimizations.

Although a limited grid of equispaced α values is selected for visualization purposes, the resulting boxplots for intermediate α values can be conceptually visualized as being placed *in between* the two nearest boxplots. For example, if we consider the $n = 25$ scenario on the μ parameter of the univariate Normal random sample model (top-left subpanel of Figure 2.4), then the Power-EP approximations with $0.75 < \alpha < 1$ would assess an accuracy score between 98.5% and 99% for most of

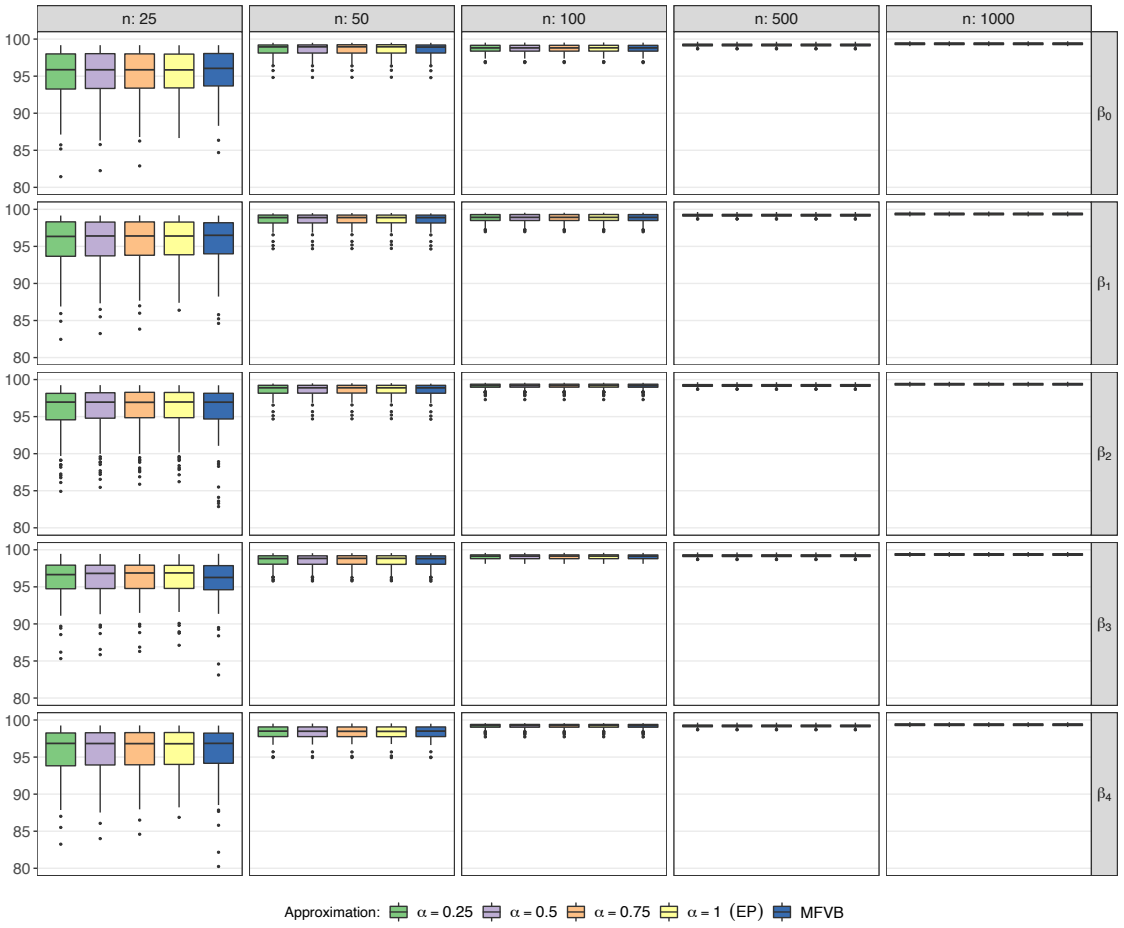

FIGURE 2.7: *Accuracy values for the optimal approximate densities of the logit regression model, obtained from the simulation study. Each row corresponds to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\beta_h)$ *and* $\mathfrak{q}^*_{MFVB}(\beta_h)$ *to* $\mathfrak{p}(\beta_h|\boldsymbol{y})$ *for* $0 \leq h \leq 4$*. Different choices of* α *are indicated in the legend.*

the simulated random data samples. Similarly, by looking at the $n = 25$ scenario on the σ^2 parameter (bottom-left subpanel of Figure 2.4), we immediately see that no Power-EP approximations with $\alpha \in (0, 1)$ lead to accuracy scores overwhelming those obtained by EP.

Finally, remember VB can be interpreted as the limiting case for Power-EP when $\alpha \to 0$. Nonetheless, the resulting approximating densities would differ from those obtained by MFVB because Power-EP adds a proper parametric family assumption for each approximating density function into which the generic $\mathfrak{q} \in \mathcal{Q}$ is factorized. For this reason, we placed MFVB accuracy boxplots in the far-right position for each sub-panel instead of the more intuitive far-left position. A numerical counterpart for the VB approximations could be obtained running our proposed Power-EP algorithms with, e.g., $\alpha = 0.05$. Nevertheless, we remark that pure VB and MFVB approximations are not included in the Power-EP approximating family: as such,

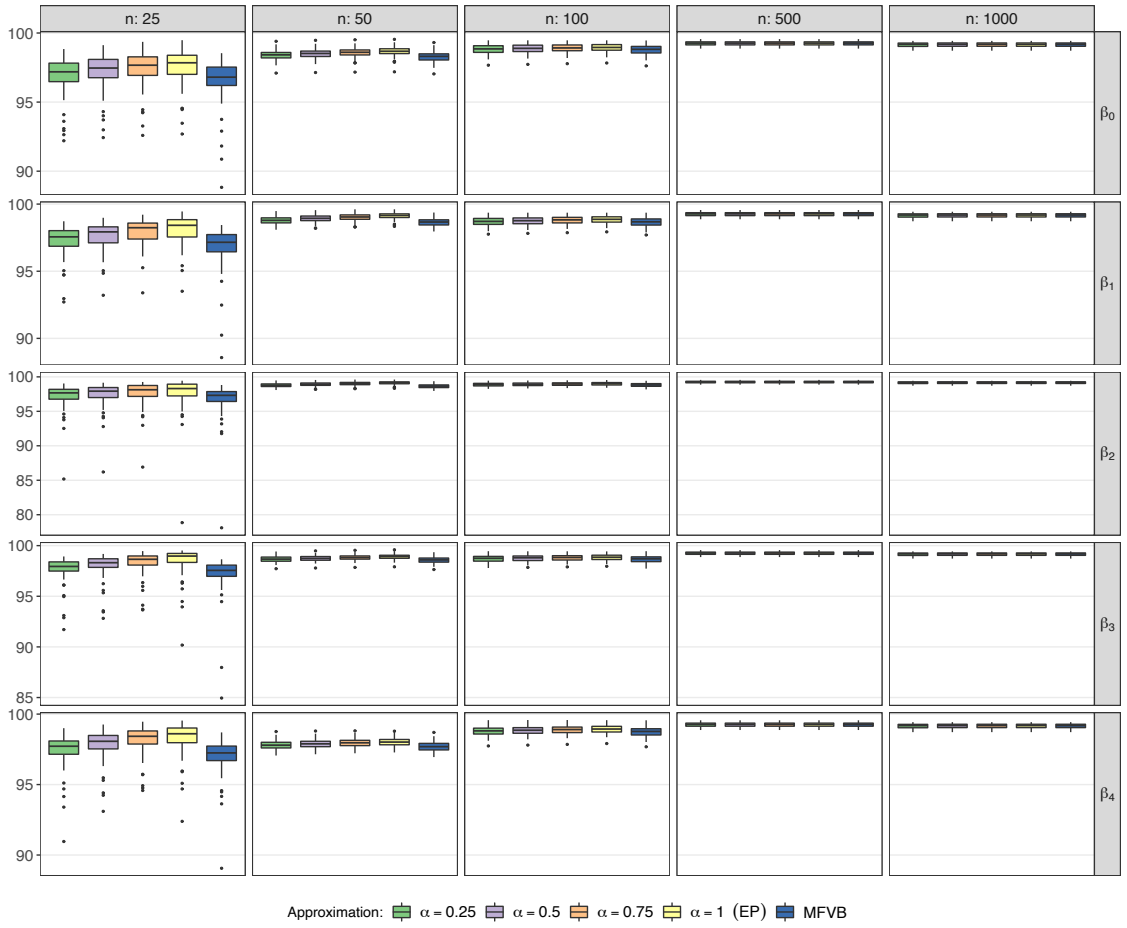

FIGURE 2.8: *Accuracy values for the optimal approximate densities of the Poisson regression model, obtained from the simulation study. Each row corresponds to the optimal approximating densities* $\mathfrak{q}^*_\alpha(\beta_h)$ *and* $\mathfrak{q}^*_{MFVB}(\beta_h)$ *to* $\mathfrak{p}(\beta_h|\boldsymbol{y})$ *for* $0 \leq h \leq 4$*. Different choices of* α *are indicated in the legend.*

they must be treated as limiting competitive approximations on which to compare Power-EP approximations.

One final comment involves the computational timings for all the approximating methods considered. As already discussed in Section 1.5, a genuine comparison between the convergence time of the algorithms for Power-EP approximations, MFVB approximations and MCMC sampling is not possible. Nevertheless, our simulation study confirmed what is already known from the literature: Power-EP and MFVB algorithms converged in significantly faster timings than those required for a successfully-converged MCMC sampler. Among them, MFVB algorithms were faster than those implementing Power-EP approximations because they do not involve any numerical integrations to be computed. Regarding the Power-EP family of approximations, we have noticed that the convergence is reached with a number of iterations that increases as $\alpha \to 0$, making EP algorithm always the fastest.

The reason is imputable again to the Power-EP natural parameter vector updates structure. In fact, as $\alpha \to 0$ it assigns a way more relevant weight on the additional contribution to its value computed in the previous iteration. Although this ensures a more robust strategy for reaching convergence, it requires more iterations to be performed.

2.7 Concluding Remarks

Power-EP approximations are overshadowed by standard EP and MFVB variational methods, especially in pure statistical contexts, and Kullback-Leibler divergence minimization still constitutes the main optimization problem faced by variational methods. The work presented in this chapter is a novel attempt to investigate and develop explicit algorithms for obtaining variational approximations from the α-divergence family. Their application on some of the most common statistical models allowed us to derive explicit algebraic computations for implementing them in standard statistical software. Such derivations could be useful for future work, expanding towards more complex models. Sources of inspiration have been the seminal works on EP approximations from a statistical perspective by Kim and Wand (2016), Kim and Wand (2018) and Chen and Wand (2020). They motivated us to examine if their EP derivations could be easily extended to account for Power-EP approximations and, if so, which improvements could emerge.

Our empirical investigations have found few advantages that may suggest the usage of Power-EP for obtaining variational approximations in all the statistical models considered. Reasons are imputable to the fact that, for all of them, MFVB and EP still achieve optimal results both in terms of accuracy and computational runtimes, which are rarely improvable. Therefore, accounting for alternative divergence measures to be minimized turns out to be useless in practical applications.

Nonetheless, we believe some exciting results have emerged from our research work. Firstly, algorithms for obtaining EP approximations have been embedded into those for obtaining Power-EP approximations. As such, EP appears to be one notable case of interest among an extensive set of alternative variational approximations to be found within the same algorithmic structure. The effort required for deriving a unifying algorithmic structure for Power-EP approximations allowed us to propose strategies alternative to the *derived variable trick* proposed in Kim and Wand (2018) for facing multivariate Normal projections, leading to more concise

model specifications not requiring the introduction of additional augmented variables. Moreover, we have derived explicit natural parameter vector update expressions for doing EP approximations on some notable GLMs.

Secondly, the idea of exploring approximations conceptually lying between VB and EP is certainly not exhaustively examined in variational literature, especially from a statistical viewpoint. Recasting them as boundary solutions for the $\alpha \in (0, 1]$ interval is more than a philosophical exercise. Indeed, it allows us to examine all the variational approximations arising between them and to comment whether they can improve or worsen EP and VB approximations. Empirical experiments allowed us to show that some minor improvements arise for some model parameters, although within low sample size settings. Further work into developing Power-EP algorithms for more complicated statistical models may be necessary to understand whether α-divergences could effectively provide a more general geometrical perspective on measuring the distance between the model posterior distribution and the approximating one if compared to the Kullback-Leibler divergence. Moreover, it would also be interesting to examine the quality of approximations found within Power-EP selecting $\alpha < 0$ or $\alpha > 1$, thus exploring beyond VB and EP.

Regarding the choice of which α yields the best Power-EP approximation, this is still an open question in variational literature and different answers are possible, depending on the statistical model considered, on the observed data, and on the approximation required. One possible approach is that of Hernandez-Lobato *et al.* (2016), evaluating accuracy metrics over a test set, although this technique is only possible for machine learning models. Alternatively one can start selecting a grid of possible competing α values, obtaining their Power-EP approximations and computing the associated lower bounds having expression (2.6). The search can then be further restricted to a thicker grid of α values selected among those exhibiting the maximum values of the lower bounds.

Chapter 3

Fixed Effects Selection for Multilevel Models via Streamlined MFVB

3.1 Introduction

Various statistical models can be formulated as linear regression models incorporating both fixed and random effects in the linear predictor. The former are effects associated with the entire population or repeatable levels of experimental factors; the latter arise from individual experimental units drawn at random. In the statistical literature, models admitting both fixed and random effects are known as mixed-effects models (Pinheiro and Bates, 2006). These models are employed in an assortment of regression-type studies, including the analysis of classical longitudinal data (e.g. Fitzmaurice *et al.*, 2008), repeated measurements (e.g. Vonesh and Chinchilli, 1997), blocked designs (e.g. Lindner and Rodger, 2009), multilevel data (e.g. Goldstein, 2010), as well as semi-parametric regression models (e.g. Ruppert *et al.*, 2003) such as those including spatial or spline-type components.

The focus of this chapter is on Bayesian fitting of linear mixed-effects models with nested random effects structures, which are commonly used for the analysis of longitudinal, multilevel, and panel data (e.g. Verbeke and Molenberghs, 2000; Baltagi, 2021), or small area estimation (e.g. Rao and Molina, 2015). These data are typically collected from experimental units that can be grouped into different levels of nesting, and the interest is in modeling within-group correlations.

In areas of application such as genome-wide association studies (e.g. Korte *et al.*, 2012; Sikorska *et al.*, 2013; Li *et al.*, 2015a) and medical research (e.g. Brown and Prescott, 2015), datasets typically possess a large number of group-invariant predictors, of which only a few are effectively relevant. A common misleading strategy

is that of including all of them as fixed effects in the model specification. This may compromise the parsimony of the model specification and validity of inferential conclusions, especially in sparse covariates settings. Therefore, a properly fixed effects selection procedure is recommended to identify those being effectively relevant. Although many frequentist procedures have been developed to tackle this problem (e.g. Schelldorfer *et al.*, 2011; Fan and Li, 2012; Groll and Tutz, 2014; Hui *et al.*, 2017; Li *et al.*, 2018), little exists in the Bayesian literature. Bayesian approaches are mostly focused on random effects selection induced by the decomposition of their covariance matrix (e.g. Chen and Dunson, 2003; Yang, 2013), or joint fixed and random effects selection (e.g. Kinney and Dunson, 2007; Yang *et al.*, 2020).

The current chapter focuses on fixed effects selection procedures from a Bayesian perspective. This may be advantageous over frequentist approaches, especially in high-dimensional settings when likelihood-based inference is computationally intractable and allows prior knowledge about the parameters to be incorporated into the model specification. MCMC sampling still represents the reference toolkit for *exact* Bayesian inference, and all the references mentioned above on Bayesian approaches for effects selection perform model fitting via MCMC. However, such a procedure usually generates higher computational times and, in general, necessitates convergence assessment for all the model parameters, which may cause problems such as poor mixing connected to the model parameterization. These and other drawbacks have supported the development of variational approximations for linear mixed models to improve convergence speed at the cost of employing an approximation of the true posterior distribution.

Wang and Wand (2011) provided some insights on how to implement variational approximations for approximate Bayesian inference in hierarchical models through Infer.NET (Minka *et al.*, 2018). Although this computational framework is suitable for longitudinal and multilevel models, its computational advantage quickly decreases for high numbers of groups and sub-groups, limiting the usefulness of variational inference. Algorithm 3 of Ormerod and Wand (2010), and Algorithms 3 and 5 of Luts *et al.* (2014) allow to implement variational inference for fitting longitudinal and multilevel data; however, they do not perform efficiently for large dimensions, as they include naïve updates based on inefficient matrix inversions.

Lee and Wand (2016) developed a streamlined updating scheme for variational inference making efficient use of sub-matrix inversion operations whose number is linear in the size of groups at each level. The streamlined scheme represents an improvement of two orders of magnitude over naïve implementations of variational

approximations. These results have also been extended to the class of generalized linear mixed-effects models and applied, for instance, to models for multiple longitudinal markers (Hughes *et al.*, 2021). Nolan *et al.* (2020) took advantage of the sparse matrix results developed in Nolan and Wand (2020) for deriving efficient algorithms and performing efficient Bayesian variational approximations for linear mixed models with two and three-level nested random effects structures. This framework, named *streamlined variational inference*, allows to dramatically reduce computational times when compared to naïve implementations of variational inference, although achieving the same approximation. Furthermore, streamlined variational inference efficiently stores the matrices needed to perform algorithm updates, hence providing significant memory savings.

These developments have recently inspired streamlined algorithms for linear mixed-effects models with crossed random effects (Menictas *et al.*, 2019) and group-specific curves (Menictas *et al.*, 2021). Many extensions can be envisaged and are motivated by the high demand for fast and accurate processing methods for big amounts of data from clinical studies, psychological experiments, or surveys in social sciences. Nonetheless, the current streamlined variational algorithms have been developed and tested using generic uninformative Normal priors over the fixed effects vector. In this chapter, we introduce streamlined variational inference for models with global-local priors inducing Bayesian posterior shrinkage and we study an efficient selection procedure for fixed effects.

To the best of our knowledge, the literature lacks of scalable variational approximation methods for fixed effects selection in linear mixed models. Armagan and Dunson (2011) proposed a sparse variational Bayes analysis of linear mixed models, which focuses on random effects shrinkage via decomposition of the random effects vector covariance matrix. A more recent contribution is Tung *et al.* (2019), where the suggested approach performs simultaneous fixed-effect selection and parameter estimation via variational Bayes and Bayesian adaptive lasso. However, their approach is limited to high-dimensional two-level generalized linear mixed models and does not account for any streamlined updating improvements.

The work presented in this chapter extends the results and algorithms of Nolan *et al.* (2020) by developing streamlined Bayesian variational approximations for multilevel linear mixed models with two or three-level random effects where a subset of fixed effects is subject to selection. The selection is performed by first placing global-local priors over the fixed effects being subject to selection, which ensures good shrinkage properties towards the origin for irrelevant fixed effects marginal

posteriors, and then identifying those being relevant via an automated selection procedure free from hyperparameters tuning.

3.2 Linear Mixed Models

This chapter treats linear mixed models with Gaussian responses and homoskedastic independent errors. A general specification for these models is:

$$\begin{gathered}
\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{u},\sigma^2 \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{u},\sigma^2\boldsymbol{I}), \quad \boldsymbol{u}|\boldsymbol{G} \sim \mathrm{N}(\boldsymbol{0},\boldsymbol{G}), \\
\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}},\boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \\
\sigma^2|a_{\sigma^2} \sim \text{Inverse-}\chi^2(\nu_{\sigma^2},1/a_{\sigma^2}), \quad a_{\sigma^2} \sim \text{Inverse-}\chi^2(1,1/(\nu_{\sigma^2}s^2_{\sigma^2})), \\
\boldsymbol{G}|\boldsymbol{A}_G \sim \mathfrak{p}(\boldsymbol{G}|\boldsymbol{A}_G), \quad \boldsymbol{A}_G \sim \mathfrak{p}(\boldsymbol{A}_G),
\end{gathered} \tag{3.1}$$

where $\boldsymbol{y}$ is a vector of observed data, $\boldsymbol{\beta}$ and $\boldsymbol{u}$ are the vectors of fixed and random effects, $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the associated fixed and random effects design matrices, σ^2 is the variance of the unit-specific error term, and $\boldsymbol{G}$ is the random effects covariance matrix.

A very general prior specification for the parameters of model (3.1) is typically considered. The fixed effects vector $\boldsymbol{\beta}$ has a multivariate Normal prior with hyperparameters $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$. Following Gelman (2006), the hierarchical prior specification on σ^2 generates a Half-t distribution on σ with ν_{σ^2} degrees of freedom and scale parameter s_{σ^2}, where larger values of s_{σ^2} correspond to weaker informativity. A similar hierarchical prior is imposed on the random effects vector covariance matrix $\boldsymbol{G}$: if, for instance, $\mathfrak{p}(\boldsymbol{G}|\boldsymbol{A}_G)$ is an Inverse-G-Wishart$(G_{\text{full}},\nu_G+2q-2,\boldsymbol{A}_G^{-1})$ density function and $\mathfrak{p}(\boldsymbol{A}_G)$ is an Inverse-G-Wishart$(G_{\text{diag}},1,\boldsymbol{\Lambda}_{\boldsymbol{A}_G})$ density function with $\boldsymbol{\Lambda}_{\boldsymbol{A}_G} \equiv \{\nu_G\,\text{diag}(s^2_{G,1},\ldots,s^2_{G,q})\}^{-1}$, then according to Huang and Wand (2013) such a prior imposition may induce arbitrarily noninformative priors on the standard deviation parameters for large values of $s_{G,1},\ldots,s_{G,q}$ and a Uniform$(-1,1)$ distribution over the correlation parameters. Details are provided in Appendix A.2.

The structures of $\boldsymbol{Z}$, $\boldsymbol{u}$, and $\boldsymbol{G}$ embed a rich ensemble of mixed model specifications, as shown by Zhao *et al.* (2006). In this chapter, we will focus on multilevel models having two-level and three-level random effects specifications.

3.2.1 Two-Level Linear Mixed Models

Multilevel models with two-level random effects arise from applications where observations from different units belonging to separate groups are available, and the interest is in capturing the within-group variability. Let m be the number of groups, each composed by o_i units, $1 \leq i \leq m$. A two-level linear mixed model can be expressed in terms of the observations from the ith group as follows:

$$\begin{gathered}
\boldsymbol{y}_i|\boldsymbol{\beta}, \boldsymbol{u}_i, \sigma^2 \overset{\text{ind}}{\sim} \mathrm{N}(\boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{u}_i, \sigma^2\boldsymbol{I}), \quad \boldsymbol{u}_i|\boldsymbol{\Sigma} \overset{\text{ind}}{\sim} \mathrm{N}(\boldsymbol{0}, \boldsymbol{\Sigma}), \quad 1 \leq i \leq m, \\
\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \\
\sigma^2|a_{\sigma^2} \sim \text{Inverse-}\chi^2(\nu_{\sigma^2}, 1/a_{\sigma^2}), \quad a_{\sigma^2} \sim \text{Inverse-}\chi^2(1, 1/(\nu_{\sigma^2}s^2_{\sigma^2})), \\
\boldsymbol{\Sigma}|\boldsymbol{A}_{\boldsymbol{\Sigma}} \sim \text{Inverse-G-Wishart}(G_{\text{full}}, \nu_{\boldsymbol{\Sigma}} + 2q - 2, \boldsymbol{A}_{\boldsymbol{\Sigma}}^{-1}), \\
\boldsymbol{A}_{\boldsymbol{\Sigma}} \sim \text{Inverse-G-Wishart}(G_{\text{diag}}, 1, \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}}})
\end{gathered} \tag{3.2}$$

with $\boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}}} \equiv \{\nu_{\boldsymbol{\Sigma}}\,\mathrm{diag}(s^2_{\boldsymbol{\Sigma},1}, \ldots, s^2_{\boldsymbol{\Sigma},q})\}^{-1}$. Here $\boldsymbol{\Sigma}$ is the covariance matrix for the group-specific random effects vector $\boldsymbol{u}_i$ of length q. Notice model (3.2) is a particular case of model (3.1), with

$$\begin{gathered}
\boldsymbol{y} = \underset{1\leq i\leq m}{\mathrm{stack}}(\boldsymbol{y}_i), \quad \boldsymbol{X} = \underset{1\leq i\leq m}{\mathrm{stack}}(\boldsymbol{X}_i), \quad \boldsymbol{Z} = \underset{1\leq i\leq m}{\mathrm{blockdiag}}(\boldsymbol{Z}_i), \\
\boldsymbol{u} = \underset{1\leq i\leq m}{\mathrm{stack}}(\boldsymbol{u}_i), \quad \boldsymbol{G} = \boldsymbol{I}_m \otimes \boldsymbol{\Sigma}, \quad \boldsymbol{A}_G = \boldsymbol{I}_m \otimes \boldsymbol{A}_{\boldsymbol{\Sigma}}.
\end{gathered} \tag{3.3}$$

The structure of $\boldsymbol{Z}$ is such that $\boldsymbol{Z}\boldsymbol{u} = \sum_{i=1}^{m} \boldsymbol{Z}_i\boldsymbol{u}_i$ and notice that as m increases, $\boldsymbol{Z}$ becomes sparser with only the $(100/m)\%$ of its cells being non-zero.

3.2.2 Three-Level Linear Mixed Models

Multilevel models with three-level random effects extend the two-level specification by adding a further hierarchy level. Such structures are employed when there is interest in capturing the variability within groups and that within their subgroups.

Let m denote the number of level 1 (L1) groups, n_i be the number of level 2 (L2) subgroups belonging to the ith group, $1 \leq i \leq m$, and o_{ij} be the number of units belonging to the jth subgroup, $1 \leq j \leq n_i$, of the ith group. A three-level linear mixed model can be defined in terms of the observations from the jth subgroup

belonging to the ith group as follows:

$$
\begin{aligned}
\boldsymbol{y}_{ij}|\boldsymbol{\beta}, \boldsymbol{u}_i^{\text{L1}}, \boldsymbol{u}_{ij}^{\text{L2}}, \sigma^2 &\overset{\text{ind}}{\sim} \text{N}(\boldsymbol{X}_{ij}\boldsymbol{\beta} + \boldsymbol{Z}_{ij}^{\text{L1}}\boldsymbol{u}_i^{\text{L1}} + \boldsymbol{Z}_{ij}^{\text{L2}}\boldsymbol{u}_{ij}^{\text{L2}}, \sigma^2\boldsymbol{I}), \\
\left[\begin{array}{c} \boldsymbol{u}_i^{\text{L1}} \\ \boldsymbol{u}_{ij}^{\text{L2}} \end{array}\right] \Big| \, \boldsymbol{\Sigma}^{\text{L1}}, \boldsymbol{\Sigma}^{\text{L2}} &\overset{\text{ind}}{\sim} \text{N}\left(\left[\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{0} \end{array}\right], \left[\begin{array}{cc} \boldsymbol{\Sigma}^{\text{L1}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Sigma}^{\text{L2}} \end{array}\right]\right), \quad 1 \leq i \leq m,\, 1 \leq j \leq n_i, \\
\boldsymbol{\beta} &\sim \text{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}}), \\
\sigma^2|a_{\sigma^2} \sim \text{Inverse-}\chi^2(\nu_{\sigma^2}, 1/a_{\sigma^2}), \quad a_{\sigma^2} &\sim \text{Inverse-}\chi^2(1, 1/(\nu_{\sigma^2}s_{\sigma^2}^2)), \\
\boldsymbol{\Sigma}^{\text{L1}}|\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}} &\sim \text{Inverse-G-Wishart}(G_{\text{full}}, \nu_{\boldsymbol{\Sigma}^{\text{L1}}} + 2q_1 - 2, \boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}}^{-1}), \\
\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}} &\sim \text{Inverse-G-Wishart}(G_{\text{diag}}, 1, \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}}}), \\
\boldsymbol{\Sigma}^{\text{L2}}|\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}} &\sim \text{Inverse-G-Wishart}(G_{\text{full}}, \nu_{\boldsymbol{\Sigma}^{\text{L2}}} + 2q_2 - 2, \boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}}^{-1}), \\
\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}} &\sim \text{Inverse-G-Wishart}(G_{\text{diag}}, 1, \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}}})
\end{aligned}
\tag{3.4}
$$

with $\boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}}} \equiv \{\nu_{\boldsymbol{\Sigma}^{\text{L1}}} \, \text{diag}(s^2_{\boldsymbol{\Sigma}^{\text{L1}},1}, \ldots, s^2_{\boldsymbol{\Sigma}^{\text{L1}},q_1})\}^{-1}$ and $\boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}}} \equiv \{\nu_{\boldsymbol{\Sigma}^{\text{L2}}} \, \text{diag}(s^2_{\boldsymbol{\Sigma}^{\text{L2}},1}, \ldots, s^2_{\boldsymbol{\Sigma}^{\text{L2}},q_2}),\}^{-1}$. Here $\boldsymbol{\Sigma}^{\text{L1}}$ is the covariance matrix for the group-specific random effects vector $\boldsymbol{u}_i^{\text{L1}}$ of length q_1 and $\boldsymbol{\Sigma}^{\text{L2}}$ is that for the subgroup-specific random effects vector $\boldsymbol{u}_{ij}^{\text{L2}}$ having length q_2. Notice model (3.4) is a particular case of model (3.1), with

$$
\begin{aligned}
\boldsymbol{y} &= \underset{1\leq i\leq m}{\text{stack}} \left(\underset{1\leq j\leq n_i}{\text{stack}} (\boldsymbol{y}_{ij})\right), \quad \boldsymbol{X} = \underset{1\leq i\leq m}{\text{stack}} \left(\underset{1\leq j\leq n_i}{\text{stack}} (\boldsymbol{X}_{ij})\right), \\
\boldsymbol{Z} &= \underset{1\leq i\leq m}{\text{blockdiag}} \left(\left[\begin{array}{c|c} \underset{1\leq j\leq n_i}{\text{stack}} (\boldsymbol{Z}_{ij}^{\text{L1}}) & \underset{1\leq j\leq n_i}{\text{blockdiag}} (\boldsymbol{Z}_{ij}^{\text{L2}}) \end{array}\right]\right), \\
\boldsymbol{u} &= \underset{1\leq i\leq m}{\text{stack}} \left(\left[\begin{array}{c|c} (\boldsymbol{u}_i^{\text{L1}})^T & \left(\underset{1\leq j\leq n_i}{\text{stack}} (\boldsymbol{u}_{ij}^{\text{L2}})\right)^T \end{array}\right]^T\right), \\
\boldsymbol{G} &= \underset{1\leq i\leq m}{\text{blockdiag}} \left(\left[\begin{array}{cc} \boldsymbol{\Sigma}^{\text{L1}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n_i} \otimes \boldsymbol{\Sigma}^{\text{L2}} \end{array}\right]\right), \\
\boldsymbol{A}_G &= \underset{1\leq i\leq m}{\text{blockdiag}} \left(\left[\begin{array}{cc} \boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n_i} \otimes \boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}} \end{array}\right]\right).
\end{aligned}
\tag{3.5}
$$

The structure of $\boldsymbol{Z}$ is more involved than the one of two-level random effects models and is such that $\boldsymbol{Z}\boldsymbol{u} = \sum_{i=1}^{m}\sum_{j=1}^{n_i}(\boldsymbol{Z}_{ij}^{\text{L1}}\boldsymbol{u}_i^{\text{L1}} + \boldsymbol{Z}_{ij}^{\text{L2}}\boldsymbol{u}_{ij}^{\text{L2}})$. As m and the n_i's increase, $\boldsymbol{Z}$ becomes sparser with only the $\{(q_1+q_2)/(q_1 m + q_2\sum_{i=1}^{m} n_i) \times 100\}\%$ of its cells being non-zero.

Notice that in the particular case where $n_i = 1$ for all $1 \leq i \leq m$ and $q_1 = q_2 = q$, the three-level specification corresponds to the two-level one.

3.3 Mean Field Variational Bayes Approximations

We now give technical details on MFVB approximations for fitting two- and three-level linear mixed models. We present the so-called naïve MFVB updates resulting from the straightforward implementation of standard formulas from Section 1.2, together with the streamlined MFVB idea providing a more efficient treatment of sparse matrix structures arising within such updates.

Assume then the posterior density function $\mathfrak{p}(\boldsymbol{\beta}, \boldsymbol{u}, \sigma^2, a_{\sigma^2}, \boldsymbol{G}, \boldsymbol{A}_G|\boldsymbol{y})$ of the generic linear mixed model specification (3.1) is approximated via (1.2) by a generic $\mathfrak{q}$-density function being factorized as follows:

$$\mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u}, \sigma^2, a_{\sigma^2}, \boldsymbol{G}, \boldsymbol{A}_G) = \mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u}, a_{\sigma^2}, \boldsymbol{A}_G)\, \mathfrak{q}(\sigma^2, \boldsymbol{G}).$$

This factorization represents the minimal product restriction for which practical variational inference algorithms arise, see Section 2 of Nolan *et al.* (2020). However, the interaction between the partition assumed for the global parameter vector and the conditional independence properties underlying model specification (3.1) admit the following *induced factorization* for the generic $\mathfrak{q} \in \mathcal{Q}$, namely:

$$\mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u}, \sigma^2, a_{\sigma^2}, \boldsymbol{G}, \boldsymbol{A}_G) = \mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u})\, \mathfrak{q}(\sigma^2)\, \mathfrak{q}(a_{\sigma^2})\, \mathfrak{q}(\boldsymbol{G})\, \mathfrak{q}(\boldsymbol{A}_G).$$

Using arguments from Section 1.2, it is possible to show that the optimal approximating densities are:

$$\begin{aligned} \mathfrak{q}^*(\boldsymbol{\beta}, \boldsymbol{u}) &\text{ is a } \mathrm{N}\big(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}\big) \text{ density function ,} \\ \mathfrak{q}^*(\sigma^2) &\text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(\sigma^2)}, \lambda_{\mathfrak{q}(\sigma^2)}\big) \text{ density function ,} \\ \mathfrak{q}^*(a_{\sigma^2}) &\text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(a_{\sigma^2})}, \lambda_{\mathfrak{q}(a_{\sigma^2})}\big) \text{ density function .} \end{aligned} \tag{3.6}$$

The approximating densities for the matrices $\boldsymbol{G}$ and $\boldsymbol{A}_G$ vary according to the random effects structure considered and are related to the structure of the random effects design matrix. For the two-level random effects specification, the structures of $\boldsymbol{G}$ and $\boldsymbol{A}_G$ are given by the second row of (3.3), which involves matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{A}_\Sigma$. Therefore, their approximating densities are:

$$\begin{aligned} \mathfrak{q}^*(\boldsymbol{\Sigma}) &\text{ is an Inverse-G-Wishart}\big(G_{\text{full}}, \xi_{\mathfrak{q}(\boldsymbol{\Sigma})}, \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{\Sigma})}\big) \text{ density function ,} \\ \mathfrak{q}^*(\boldsymbol{A}_\Sigma) &\text{ is an Inverse-G-Wishart}\big(G_{\text{diag}}, \xi_{\mathfrak{q}(\boldsymbol{A}_\Sigma)}, \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_\Sigma)}\big) \text{ density function .} \end{aligned}$$

For the three-level random effects specification, the structures of $\boldsymbol{G}$ and $\boldsymbol{A}_G$ are given by the last two rows of (3.5), which involve matrices $\boldsymbol{\Sigma}^{\text{L1}}$, $\boldsymbol{\Sigma}^{\text{L2}}$, $\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}}$ and $\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}}$. Therefore, their approximating densities are:

$$\begin{aligned}
\mathfrak{q}^*(\boldsymbol{\Sigma}^{\text{L1}}) &\text{ is an Inverse-G-Wishart}\big(G_{\text{full}}, \xi_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L1}})}, \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L1}})}\big) \text{ density function ,}\\
\mathfrak{q}^*(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}}) &\text{ is an Inverse-G-Wishart}\big(G_{\text{diag}}, \xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})}, \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})}\big) \text{ density function ,}\\
\mathfrak{q}^*(\boldsymbol{\Sigma}^{\text{L2}}) &\text{ is an Inverse-G-Wishart}\big(G_{\text{full}}, \xi_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L2}})}, \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L2}})}\big) \text{ density function ,}\\
\mathfrak{q}^*(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}}) &\text{ is an Inverse-G-Wishart}\big(G_{\text{diag}}, \xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})}, \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})}\big) \text{ density function .}
\end{aligned}$$

3.3.1 Naïve MFVB Updates

Let $\boldsymbol{C} = [\,\boldsymbol{X} \,|\, \boldsymbol{Z}\,]$ and denote with n the number of its rows. The parameters of these optimal $\mathfrak{q}$-densities can be obtained by iteratively performing the following updates until convergence:

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} &\longleftarrow \left(\mu_{\mathfrak{q}(1/\sigma^2)}\, \boldsymbol{C}^T\boldsymbol{C} + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & E_{\mathfrak{q}}(\boldsymbol{G}^{-1}) \end{bmatrix}\right)^{-1},\\
\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} &\longleftarrow \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}\left(\mu_{\mathfrak{q}(1/\sigma^2)}\, \boldsymbol{C}^T\boldsymbol{y} + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ \boldsymbol{0} \end{bmatrix}\right),
\end{aligned} \tag{3.7}$$

$$\begin{gathered}
\xi_{\mathfrak{q}(\sigma^2)} \longleftarrow \nu_{\sigma^2} + n,\ \lambda_{\mathfrak{q}(\sigma^2)} \longleftarrow \mu_{\mathfrak{q}(1/a_{\sigma^2})} + \|\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}\|^2 + \operatorname{tr}\left\{\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}\boldsymbol{C}^T\boldsymbol{C}\right\},\\
\xi_{\mathfrak{q}(a_{\sigma^2})} \longleftarrow \nu_{\sigma^2} + 1 \quad\text{and}\quad \lambda_{\mathfrak{q}(a_{\sigma^2})} \longleftarrow \mu_{\mathfrak{q}(1/\sigma^2)} + 1/(\nu_{\sigma^2} s_{\sigma^2}^2).
\end{gathered}$$

Expressions for updating $\mu_{\mathfrak{q}(1/\sigma^2)}$ and $\mu_{\mathfrak{q}(1/a_{\sigma^2})}$ are $\mu_{\mathfrak{q}(1/\sigma^2)} \longleftarrow \xi_{\mathfrak{q}(\sigma^2)}/\lambda_{\mathfrak{q}(\sigma^2)}$ and $\mu_{\mathfrak{q}(1/a_{\sigma^2})} \longleftarrow \xi_{\mathfrak{q}(a_{\sigma^2})}/\lambda_{\mathfrak{q}(a_{\sigma^2})}$, respectively.

For a two-level random effects specification, the optimal approximating densities $\mathfrak{q}^*(\boldsymbol{\Sigma})$ and $\mathfrak{q}^*(\boldsymbol{A}_{\boldsymbol{\Sigma}})$ can be obtained by iteratively performing the following updates:

$$\begin{gathered}
\xi_{\mathfrak{q}(\boldsymbol{\Sigma})} \longleftarrow \nu_{\boldsymbol{\Sigma}} + m + 2q - 2, \quad \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{\Sigma})} \longleftarrow \boldsymbol{M}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}}^{-1})} + \sum_{i=1}^{m}\left\{\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i)}\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i)}^T + \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_i)}\right\},\\
\xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}})} \longleftarrow \nu_{\boldsymbol{\Sigma}} + q \quad\text{and}\quad \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}})} \longleftarrow \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}}} + \operatorname{diag}\left\{\text{diagonal}\left(\boldsymbol{M}_{\mathfrak{q}(\boldsymbol{\Sigma}^{-1})}\right)\right\},
\end{gathered}$$

where $\boldsymbol{M}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}}^{-1})} \longleftarrow \xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}})}\boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}})}^{-1}$, and $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i)}$ and $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_i)}$ respectively correspond to the sub-vector of $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ and sub-matrix of $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ associated with the ith group random effects vector $\boldsymbol{u}_i$, for $1 \le i \le m$. Also, the update for $E_{\mathfrak{q}}(\boldsymbol{G}^{-1})$ appearing in the first line of (3.7) is

$$E_{\mathfrak{q}}(\boldsymbol{G}^{-1}) \longleftarrow \boldsymbol{I}_m \otimes \boldsymbol{M}_{\mathfrak{q}(\boldsymbol{\Sigma}^{-1})}, \tag{3.8}$$

with $\boldsymbol{M}_{\mathfrak{q}(\boldsymbol{\Sigma}^{-1})} \longleftarrow (\xi_{\mathfrak{q}(\boldsymbol{\Sigma})} - q + 1)\boldsymbol{\Lambda}^{-1}_{\mathfrak{q}(\boldsymbol{\Sigma})}$.

For the three-level random effects specification, the optimal approximating densities $\mathfrak{q}^*(\boldsymbol{\Sigma}^{\text{L1}})$, $\mathfrak{q}^*(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})$, $\mathfrak{q}^*(\boldsymbol{\Sigma}^{\text{L2}})$ and $\mathfrak{q}^*(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})$ can be obtained by iteratively performing the following updates:

$$\xi_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L1}})} \longleftarrow \nu_{\boldsymbol{\Sigma}^{\text{L1}}} + m + 2q_1 - 2, \quad \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L1}})} \longleftarrow \boldsymbol{M}_{\mathfrak{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\text{L1}}})} + \sum_{i=1}^{m}\left\{\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i^{\text{L1}})}\boldsymbol{\mu}^T_{\mathfrak{q}(\boldsymbol{u}_i^{\text{L1}})} + \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_i^{\text{L1}})}\right\},$$

$$\xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})} \longleftarrow \nu_{\boldsymbol{\Sigma}^{\text{L1}}} + q_1, \quad \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})} \longleftarrow \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}}} + \text{diag}\left\{\text{diagonal}\left(\boldsymbol{M}_{\mathfrak{q}((\boldsymbol{\Sigma}^{\text{L1}})^{-1})}\right)\right\},$$

$$\xi_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L2}})} \longleftarrow \nu_{\boldsymbol{\Sigma}^{\text{L2}}} + \sum_{i=1}^{m} n_i + 2q_2 - 2, \quad \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L2}})} \longleftarrow \boldsymbol{M}_{\mathfrak{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\text{L2}}})} + \sum_{i=1}^{m}\sum_{j=1}^{n_i}\left\{\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{\text{L2}})}\boldsymbol{\mu}^T_{\mathfrak{q}(\boldsymbol{u}_{ij}^{\text{L2}})} + \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{\text{L2}})}\right\},$$

$$\xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})} \longleftarrow \nu_{\boldsymbol{\Sigma}^{\text{L2}}} + q_2 \quad \text{and} \quad \boldsymbol{\Lambda}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})} \longleftarrow \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}}} + \text{diag}\left\{\text{diagonal}\left(\boldsymbol{M}_{\mathfrak{q}((\boldsymbol{\Sigma}^{\text{L2}})^{-1})}\right)\right\},$$

where $\boldsymbol{M}_{\mathfrak{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\text{L1}}})} \longleftarrow \xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})}\boldsymbol{\Lambda}^{-1}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L1}}})}$, $\boldsymbol{M}_{\mathfrak{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\text{L2}}})} \longleftarrow \xi_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})}\boldsymbol{\Lambda}^{-1}_{\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\text{L2}}})}$, $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i^{\text{L1}})}$ and $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_i^{\text{L1}})}$ respectively correspond to the sub-vector of $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ and sub-matrix of $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ associated with the ith group random effects vector at level 1 $\boldsymbol{u}_i^{\text{L1}}$, and $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{\text{L2}})}$ and $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{\text{L2}})}$ respectively correspond to the sub-vector of $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ and sub-matrix of $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ associated with the jth subgroup of the ith group random effects vector at level 2 $\boldsymbol{u}_{ij}^{\text{L2}}$, for $1 \leq i \leq m$ and $1 \leq j \leq n_i$. Furthermore, the update for $E_{\mathfrak{q}}(\boldsymbol{G}^{-1})$ appearing in the first line of (3.7) is

$$E_{\mathfrak{q}}(\boldsymbol{G}^{-1}) \longleftarrow \underset{1\leq i\leq m}{\text{blockdiag}}\left(\begin{bmatrix}\boldsymbol{M}_{\mathfrak{q}((\boldsymbol{\Sigma}^{\text{L1}})^{-1})} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{I}_{n_i}\otimes\boldsymbol{M}_{\mathfrak{q}((\boldsymbol{\Sigma}^{\text{L2}})^{-1})}\end{bmatrix}\right), \tag{3.9}$$

with $\boldsymbol{M}_{\mathfrak{q}((\boldsymbol{\Sigma}^{\text{L1}})^{-1})} \longleftarrow (\xi_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L1}})} - q_1 + 1)\boldsymbol{\Lambda}^{-1}_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L1}})}$, $\boldsymbol{M}_{\mathfrak{q}((\boldsymbol{\Sigma}^{\text{L2}})^{-1})} \longleftarrow (\xi_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L2}})} - q_2 + 1)\boldsymbol{\Lambda}^{-1}_{\mathfrak{q}(\boldsymbol{\Sigma}^{\text{L2}})}$.

The term *naïve* is used in this chapter when the MFVB updates are implemented without exploiting associated sparse matrix structures. Note, for example, that the update for $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ in (3.7) involves the inversion of a potentially massive matrix whose sparse structure is induced by those of $\boldsymbol{Z}$ and $\boldsymbol{G}$. As explained in Section 3.2, when the model dimensions increase such matrices may become highly sparse and the inversion operation can face many complications, both in terms of memory storage and computational efficiency. By taking advantage of the specific random effects structure, it is possible to perform efficient *streamlined* variational updates.

3.3.2 Streamlined MFVB Updates

The concept of streamlined variational inference for linear mixed models first appears in Lee and Wand (2016), where the sparse structure of $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ is exploited for efficiently fitting a particular version of model (3.2) via MFVB. Nolan and Wand

(2020) defined sparse matrix classes arising from two-level and three-level random effects specifications and provide efficient mathematical solutions to the associated matrix inversion problems in their Theorems 2.2, 2.3, 3.2 and 3.3. Nolan *et al.* (2020) implemented such results and develop streamlined MFVB algorithms for linear mixed models having both two-level and three-level random effects specifications.

Algorithms using streamlined updates achieve the same MFVB approximations obtained with naïve updates, yet reducing memory usage and performing algebraic steps more efficiently. The former is obtained by circumventing the need of storing the zero sub-blocks of $\boldsymbol{C}$ and the sub-blocks of $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}$ which are not needed for performing the other variational updates. The latter is achieved by computing the useful sub-blocks of $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}$ with faster lower-dimensional matrix inversions, and the updates of $\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}$ and $\lambda_{q(\sigma^2)}$ only with the non-zero sub-blocks of $\boldsymbol{C}$ and $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}$.

Excellent performances both in terms of approximation accuracy, computational time and memory saving when compared to naïve MFVB or efficient MCMC samplers are shown in Nolan *et al.* (2020), especially for large values of m. Nevertheless, this reference only treats the generic $\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ prior specification for the fixed effects vector, as given in (3.1). We develop instead streamlined variational inference procedures allowing for more general prior specifications on $\boldsymbol{\beta}$ aiding selection of fixed effects.

3.4 Approximate Variable Selection with Global-Local Shrinkage Priors

Regression modeling is often concerned with the problem of selecting an optimal subset of plausible regressors having a significant impact on explaining the variability of the response variable. This is of particular interest for sparse covariate settings, where a large set of covariates is considered but only a small portion of them is effectively relevant. We refer to O'Hara and Sillanpää (2009) and references therein for an exhaustive introductory review on variable selection procedures from a Bayesian perspective.

3.4.1 Bayesian Methods for Variable Selection

Most common Bayesian approaches involve placing suitable prior distributions over the parameters subject to selection. Approaches of this type can be essentially

subdivided into two leading families, based on the so-called *spike-and-slab* priors and *global-local* shrinkage priors.

Spike-and-slab priors are two-component mixture priors. The first prior component, the *spike*, is a point mass function at zero characterizing the noise, usually given by a Dirac delta function or a Gaussian density function having mean zero and very small variance. The second component, the *slab*, is an absolutely continuous density function representing the signal of nonzero coefficients associated with relevant covariates. The slab is usually given by Laplace or Gaussian density function and is typically centered around zero. A weight parameter taking values in the unit interval is used to balance the contribution of the two components. Although being highly appealing and allowing for separate control of the level of sparsity and the size of the signal coefficients, these priors may suffer from computational hurdles in high-dimensions.

Global-local shrinkage priors are absolutely continuous shrinkage priors that are placed on each coefficient β_h, $1 \leq h \leq H$, which is subject to selection. These priors admit the following convenient scale mixture representation (Polson and Scott, 2011), for proper choices of $\mathfrak{p}(\tau)$ and $\mathfrak{p}(\zeta_h)$:

$$\beta_h|\tau,\zeta_h \overset{\text{ind}}{\sim} \mathrm{N}(0,\tau^2/\zeta_h), \quad \tau \sim \mathfrak{p}(\tau), \quad \zeta_h \overset{\text{ind}}{\sim} \mathfrak{p}(\zeta_h), \quad 1 \leq h \leq H. \tag{3.10}$$

The global variance parameter τ^2 is common to all the coefficients and induces shrinkage towards the origin in the associated posterior density; the local variance parameter ζ_h is coefficient-specific. A more general specification including the model response error variance parameter is proposed in Bhattacharya *et al.* (2016). Depending on the distributional specifications for τ and ζ_h in (3.10), many well-known shrinkage priors arise. Examples are the Horseshoe prior of Carvalho *et al.* (2009, 2010), the Bayesian lasso of Park and Casella (2008), the Normal-Gamma prior of Griffin and Brown (2010), the Normal-Exponential-Gamma prior of Griffin and Brown (2011), the generalized double Pareto prior of Armagan *et al.* (2013), the Dirichlet-Laplace prior of Bhattacharya *et al.* (2015) and the Horseshoe+ prior of Bhadra *et al.* (2017). Longer lists are given in Table 1 of Tang *et al.* (2018) and Table 2 of Bhadra *et al.* (2019).

For spike-and-slab priors, the posterior distributions of negligible coefficients present a higher weight for the spike component: this provides a direct way to detect relevant effects, and therefore to perform the selection. For global-local priors, there is no posterior spike. The posterior density function, instead, is continuous with probability mass highly concentrated around zero, and a direct way for identifying

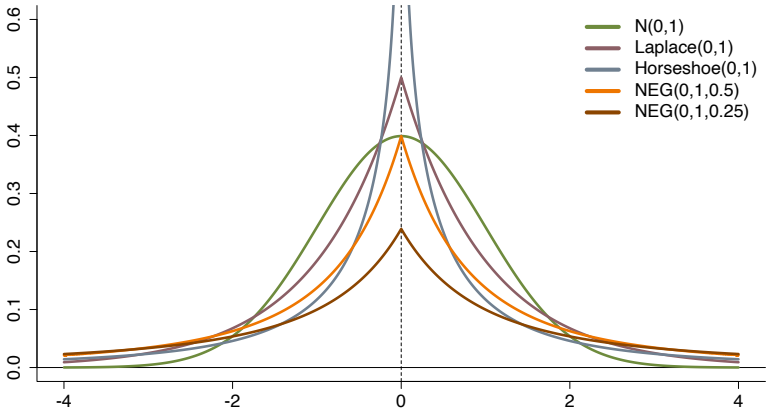

FIGURE 3.1: *Visual comparison of the probability density functions for the Laplace, Horseshoe and Normal-Exponential-Gamma (NEG) distributions with zero mean and unit standard deviation. For easiness of comparison, the standard Gaussian probability density function is also displayed.*

relevant coefficients is usually unavailable.

In this chapter we employ global-local priors as they offer substantial computational advantages over spike-and-slab priors due to their convenient representation as Gaussian scale mixtures, which give rise to convenient conjugate updates for all the β_h's and ζ_h's, and a more immediate inferential treatment. Furthermore, the estimates of frequentist regularization procedures such as ridge (Hoerl and Kennard, 1970), lasso (Tibshirani, 1996), bridge (Frank and Friedman, 1993) and elastic net (Zou and Hastie, 2005) can be recasted as posterior mode estimates from models with specific global-local priors.

3.4.2 MFVB Approximations with Global-Local Priors

Before considering their use within linear mixed model specifications, we illustrate the essential elements of mean-field variational inference for the simpler linear regression model. Without loss of generality, in this chapter we focus on three of the most commonly adopted global-local priors, namely:

$$\beta_h|\tau \overset{\text{iid}}{\sim} \text{Laplace}(0,\tau), \quad \beta_h|\tau \overset{\text{iid}}{\sim} \text{Horseshoe}(0,\tau) \quad \text{or} \quad \beta_h|\tau \overset{\text{iid}}{\sim} \text{NEG}(0,\tau,\lambda), \quad (3.11)$$

for each model coefficient β_h, $1 \leq h \leq H$. Hereafter $\lambda > 0$ is an additional shape parameter that is always assumed being user-specified.

The Laplace, Horseshoe, and Normal-Exponential-Gamma (NEG) distributions account for different degrees of prior shrinkage towards zero and have different tail behaviors, as shown in Figure 3.1. Each prior specification in (3.11) can be recasted

Prior specification	$\mathfrak{p}(\zeta_h \mid a_{\zeta_h})$	$\mathfrak{p}(a_{\zeta_h})$
Laplace$(0, \tau)$	Inverse-$\chi^2(2, 1)$	–
Horseshoe$(0, \tau)$	Gamma$(1/2, a_{\zeta_j})$	Gamma$(1/2, 1)$
NEG$(0, \tau, \lambda)$	Inverse-$\chi^2(2, 2a_{\zeta_j})$	Gamma$(\lambda, 1)$

TABLE 3.1: *Hierarchical formulation of the Laplace, Horseshoe and Negative-Exponential-Gamma (NEG) priors* (3.11) *following the general global-local representation* (3.10).

into the generic glocal-local scale mixture representation (3.10), as summarized in Table 3.1. For the Horseshoe and NEG priors, we use a convenient hierarchical representations of $\mathfrak{p}(\zeta_h)$ based on auxiliary variables a_{ζ_h}, $1 \leq h \leq H$, involving tractable Gamma and Inverse-χ^2 distributions. Details about the involved density functions and auxiliary variable representations are given in Appendix A.2.

A linear regression model with one of the three global-local priors in (3.11) is then expressible as:

$$\begin{gathered}
\boldsymbol{y} \mid \beta_0, \boldsymbol{\beta}, \sigma^2 \sim \mathrm{N}(\beta_0 \mathbf{1} + \boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}), \\
\beta_0 \sim \mathrm{N}(\mu_{\beta_0}, \sigma^2_{\beta_0}), \qquad \boldsymbol{\beta} \mid \boldsymbol{\zeta}, \tau^2 \sim \mathrm{N}(\mathbf{0}, \tau^2 \operatorname{diag}(\boldsymbol{\zeta})^{-1}), \\
\sigma^2 \mid a_{\sigma^2} \sim \text{Inverse-}\chi^2(\nu_{\sigma^2}, 1/a_{\sigma^2}), \quad a_{\sigma^2} \sim \text{Inverse-}\chi^2(1, 1/(\nu_{\sigma^2} s^2_{\sigma^2})), \\
\tau^2 \mid a_{\tau^2} \sim \text{Inverse-}\chi^2(1, 1/a_{\tau^2}), \quad a_{\tau^2} \sim \text{Inverse-}\chi^2(1, 1/s^2_{\tau^2}), \\
\zeta_h \mid a_{\zeta_h} \overset{\text{ind}}{\sim} \mathfrak{p}(\zeta_h \mid a_{\zeta_h}), \quad a_{\zeta_h} \overset{\text{ind}}{\sim} \mathfrak{p}(a_{\zeta_h}), \quad 1 \leq h \leq H,
\end{gathered} \tag{3.12}$$

where β_0 is the model intercept (which is usually excluded from the selection procedure), $\mathbf{1}$ is a vector full of ones, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_H)^T$ is the vector of coefficients having global-local shrinkage priors, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$ is the associated design matrix, $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_H)^T$ and $\boldsymbol{a_\zeta} = (a_{\zeta_1}, \ldots, a_{\zeta_H})^T$. The common Gaussian and Half-t prior distributions are considered for β_0 and σ, respectively. Importantly, we specify a Half-$t(s_{\tau^2}, 1)$ distribution with $s_{\tau^2} > 0$ for the global scale parameter τ, allowing for weak prior informativeness about the global degree of sparseness when a large scale parameter s_{τ^2} is used. Hence, we let the variable-selection procedure free from hyperparameters that must be manually tuned by the user. The densities $\mathfrak{p}(\zeta_h \mid a_{\zeta_h})$ and $\mathfrak{p}(a_{\zeta_h})$ vary according to the global-local prior specification adopted, as shown in Table 3.1.

The model posterior density function $\mathfrak{p}(\beta_0, \boldsymbol{\beta}, \sigma^2, a_{\sigma^2}, \tau^2, a_{\tau^2}, \boldsymbol{\zeta}, \boldsymbol{a}_{\boldsymbol{\zeta}} | \boldsymbol{y})$ admits a tractable MFVB approximation when the following mean-field restriction is used:

$$\mathfrak{q}(\beta_0, \boldsymbol{\beta}, \sigma^2, a_{\sigma^2}, \tau^2, a_{\tau^2}, \boldsymbol{\zeta}, \boldsymbol{a}_{\boldsymbol{\zeta}}) = \mathfrak{q}(\beta_0, \boldsymbol{\beta})\, \mathfrak{q}(\sigma^2)\, \mathfrak{q}(\tau^2)\, \mathfrak{q}(a_{\sigma^2})\, \mathfrak{q}(a_{\tau^2}) \prod_{h=1}^{H} \{\mathfrak{q}(\zeta_h)\, \mathfrak{q}(a_{\zeta_h})\}.$$

The optimal $\mathfrak{q}$-density functions then results as follows:

$$\begin{gathered}
\mathfrak{q}^*(\beta_0, \boldsymbol{\beta}) \text{ is a } \mathrm{N}\big(\boldsymbol{\mu}_{\mathfrak{q}(\beta_0, \boldsymbol{\beta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\beta_0, \boldsymbol{\beta})}\big) \text{ density function ,} \\
\mathfrak{q}^*(\sigma^2) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(\sigma^2)}, \lambda_{\mathfrak{q}(\sigma^2)}\big) \text{ density function ,} \\
\mathfrak{q}^*(a_{\sigma^2}) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(a_{\sigma^2})}, \lambda_{\mathfrak{q}(a_{\sigma^2})}\big) \text{ density function ,} \\
\mathfrak{q}^*(\tau^2) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(\tau^2)}, \lambda_{\mathfrak{q}(\tau^2)}\big) \text{ density function ,} \\
\mathfrak{q}^*(a_{\tau^2}) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(a_{\tau^2})}, \lambda_{\mathfrak{q}(a_{\tau^2})}\big) \text{ density function ,} \\
\mathfrak{q}^*(\zeta_h) \text{ is } \begin{cases} \text{an Inverse-Gaussian}(\mu_{\mathfrak{q}(\zeta_h)}, 1) \text{ density function} & \text{for a Laplace prior} \\ \text{a Gamma}(1, \lambda_{\mathfrak{q}(\zeta_h)}) \text{ density function} & \text{for a Horseshoe prior} \\ \text{an Inverse-Gaussian}(\mu_{\mathfrak{q}(\zeta_h)}, \lambda_{\mathfrak{q}(\zeta_h)}) \text{ density function} & \text{for a NEG prior ,} \end{cases} \\
\mathfrak{q}^*(a_{\zeta_h}) \text{ is } \begin{cases} - & \text{for a Laplace prior} \\ \text{a Gamma}(1, \lambda_{\mathfrak{q}(a_{\zeta_h})}) \text{ density function} & \text{for a Horseshoe prior} \\ \text{a Gamma}(\lambda + 1, \lambda_{\mathfrak{q}(a_{\zeta_h})}) \text{ density function} & \text{for a NEG prior} \end{cases}
\end{gathered} \tag{3.13}$$

for $1 \leq h \leq H$. When a Laplace prior is specified, the model does not include the a_{ζ_h} auxiliary variables and therefore $\mathfrak{q}(a_{\zeta_h})$ is not present. The expressions for the parameter updates of these approximating densities are reported in Appendix C.2, together with their full derivations.

3.4.3 From Shrinkage to Selection: the Signal Adaptive Variable Selector

One limitation of continuous global-local shrinkage priors is the unavailability of direct information from the marginal posterior of each β_h for selecting relevant coefficients. Typically, the posterior distributions of less relevant coefficients arising from such priors are highly concentrated around zero, with marked peaks and negligible tails, although not having entire mass at zero. Therefore, global-local shrinkage priors do not provide any sparse posterior solution. This issue becomes even more relevant when variational inference is used to fit the model because the

approximate marginal posterior densities of $\boldsymbol{\beta}$ are Gaussian, and so the peaks shown by the marginal posterior densities are approximated by bell-shaped curves.

Several heuristic methods have been developed for post-processing posterior distributions arising from global-local priors and determining whether the associated covariates have to be selected or not. A simple but possibly misleading solution is to select the covariates associated with coefficients whose posterior credible intervals do not contain zero. Nonetheless, this approach usually exhibits poor performances due to the difficulty of accurately estimating the uncertainty in high dimensional problems and depends on the chosen coverage level. Carvalho *et al.* (2010) defined a local shrinkage factor that can take values in the unit interval and help determine whether each variable is suggested to be selected or not according to a pre-specified threshold, analogously to the classical posterior inclusion probability of Barbieri and Berger (2004). Bondell and Reich (2012) proposed a method based on credible posterior regions, although its implementation and results rely upon the use of conjugate Normal priors. Zhang and Bondell (2018) extended the method to global-local priors and propose an intuitive approach to tune the prior hyperparameters based on minimizing a discrepancy measure between the induced distribution of R^2 from the prior and the desired distribution.

All these methods and many others share a common issue: the dependence on the choice of one or more thresholds. Bhattacharya *et al.* (2015) proposed grouping the entries of posterior medians into null and non-null groups using 2-means clustering. While this approach does not require any tuning parameters, issues emerge when signals of varying strengths are present. Li and Pati (2017) proposed a similar approach which is based on first obtaining a posterior distribution of the number of signals by clustering the signal and the noise coefficients and then estimating the signals from the resulting posterior median.

In this chapter, we opt for the signal adaptive variable selector (SAVS) partially motivated by Hahn and Carvalho (2015) and accurately developed by Ray and Bhattacharya (2018). The SAVS approach post-processes a point estimate from the posterior distribution of a coefficient having global-local prior via *soft-thresholding* to determine whether the associated covariate is assumed to be relevant or not. We adapt this procedure for usage in variational inference and propose Algorithm 3.1 as an implementation of the SAVS approach based on the optimal approximate posterior densities $\mathfrak{q}^*(\beta_h)$, $1 \leq h \leq H$. The procedure takes the approximate posterior mean parameter of a generic coefficient subject to selection and the associated unstandardized covariate as inputs. It then returns a *sparsified* approximate posterior

Algorithm 3.1 *Signal Adaptive Variable Selector (SAVS) algorithm for performing variable selection using the optimal approximate density function $\mathfrak{q}^*(\beta_h)$ of a generic coefficient with global-local prior.*

Inputs: $\mu_{\mathfrak{q}^*(\beta_h)} \equiv E_{\mathfrak{q}^*}(\beta_h)$ and $\mathbf{x}_h$ being the covariate vector corresponding to β_h.
If $\|\mathbf{x}_h\|^2 \leq |\mu_{\mathfrak{q}^*(\beta_h)}|^{-3}$:

$$\mu^*_{\mathfrak{q}^*(\beta_h)} = 0 \quad \text{and} \quad \gamma_h = 0;$$

else:

$$\mu^*_{\mathfrak{q}^*(\beta_h)} = \text{sign}(\mu_{\mathfrak{q}^*(\beta_h)})\|\mathbf{x}_h\|^{-2}\left(|\mu_{\mathfrak{q}^*(\beta_h)}|\,\|\mathbf{x}_h\|^2 - \mu^{-2}_{\mathfrak{q}^*(\beta_h)}\right) \quad \text{and} \quad \gamma_h = 1.$$

Output: A sparse estimate $\mu^*_{\mathfrak{q}^*(\beta_h)}$ for $\mathfrak{q}^*(\beta_h)$ and the associated binary selector γ_h.

summary estimate $\mu^*_{\mathfrak{q}^*(\beta_h)}$, together with a binary variable γ_h indicating whether the hth covariate is suggested to be selected or not. The attractiveness of this approach comes from the fact that it is completely automated and does not require any tuning parameters. Ray and Bhattacharya (2018) provide a theoretical justification for the SAVS approach, noticing that its output can be obtained by solving an optimization problem that is closely related to the adaptive lasso of Zou (2006), and showed it is highly competitive among alternative Bayesian selection procedures.

3.5 Linear Mixed Models with Global-Local Priors on Fixed Effects

Variational approximations for linear mixed models with two- and three-level random effects are described in Section 3.3, using a generic $\boldsymbol{\beta} \sim \text{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ prior distribution for the fixed effects parameter vector. In this work, our interest is in developing variational approximations for generalizations of model (3.1) embedding prior specifications for fixed effects selection such as those discussed in Section 3.4.

In order to do so, we subdivide the p-dimensional fixed effects vector $\boldsymbol{\beta}$ as:

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}^{\text{R}} \\ \boldsymbol{\beta}^{\text{A}} \\ \boldsymbol{\beta}^{\text{S}} \end{bmatrix},$$

where $\boldsymbol{\beta}^{\text{R}}$ is a p_{R}-dimensional vector of fixed effects associated to the random (R) effects component of the model, $\boldsymbol{\beta}^{\text{A}}$ is a p_{A}-dimensional vector of additional (A)

fixed effects and $\boldsymbol{\beta}^{\mathrm{S}}$ is a p_{S}-dimensional vector of fixed effects which are subject to selection (S). Here $p = p_{\mathrm{R}} + p_{\mathrm{A}} + p_{\mathrm{S}}$, with p_{R}, p_{A} and p_{S} varying according to the application of interest. For the two- and three-level mixed models considered in this work, $p_{\mathrm{R}} = q$ and $p_{\mathrm{R}} = \max(q_1, q_2)$ respectively. Typically, q, q_1 and q_2 are relatively small, while p_{A} and p_{S} could take moderate to large values.

Similarly, we subdivide the fixed effects design matrix as follows:

$$\boldsymbol{X} = \left[\boldsymbol{X}^{\mathrm{R}} \,\middle|\, \boldsymbol{X}^{\mathrm{A}} \,|\, \boldsymbol{X}^{\mathrm{S}} \right],$$

with $\boldsymbol{X}^{\mathrm{S}}$ assumed to have columns with zero mean and unit variance, unless differently specified. The fixed effects linear contribution in (3.1) then factorizes into $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}^{\mathrm{R}}\boldsymbol{\beta}^{\mathrm{R}} + \boldsymbol{X}^{\mathrm{A}}\boldsymbol{\beta}^{\mathrm{A}} + \boldsymbol{X}^{\mathrm{S}}\boldsymbol{\beta}^{\mathrm{S}}$, and the same applies to the two-level mixed model (3.2) and the three-level mixed model (3.4) specifications. Notice that for the latter specification, $\boldsymbol{Z}_i = \boldsymbol{X}_i^{\mathrm{R}}$ for all $1 \leq i \leq m$.

We assume without loss of generality that $\boldsymbol{\beta}^{\mathrm{R}}$, $\boldsymbol{\beta}^{\mathrm{A}}$ and $\boldsymbol{\beta}^{\mathrm{S}}$ are a-priori independent from each other, and specify the following prior distributions:

$$\boldsymbol{\beta}^{\mathrm{R}} \sim \mathrm{N}\big(\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}}\big), \quad \boldsymbol{\beta}^{\mathrm{A}} \sim \mathrm{N}\big(\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}}\big), \quad \boldsymbol{\beta}^{\mathrm{S}} \sim \mathfrak{p}(\boldsymbol{\beta}^{\mathrm{S}}) = \prod_{h=1}^{p_{\mathrm{S}}} \mathfrak{p}(\beta_h^{\mathrm{S}}),$$

with hyperparameters $\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}} \in \mathbb{R}^{p_{\mathrm{R}}}$, $\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}} \in \mathbb{R}^{p_{\mathrm{A}}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}}$ symmetric positive definite matrices. The prior specification for $\boldsymbol{\beta}^{\mathrm{S}}$ assumes a-priori independence among all the coefficients subject to selection, with $\mathfrak{p}(\beta_h^{\mathrm{S}})$ taking one of the three different global-local prior distributions treated in Section 3.4 for each $1 \leq h \leq p_{\mathrm{S}}$:

$$\beta_h^{\mathrm{S}}|\tau \overset{\text{iid}}{\sim} \text{Laplace}(0, \tau), \quad \beta_h^{\mathrm{S}}|\tau \overset{\text{iid}}{\sim} \text{Horseshoe}(0, \tau) \quad \text{or} \quad \beta_h^{\mathrm{S}}|\tau \overset{\text{iid}}{\sim} \text{NEG}(0, \tau, \lambda).$$

The resulting linear mixed model is a generalization of (3.1) that accounts for global-local prior specification over a subset of the fixed effects, and can be expressed as:

$$
\begin{aligned}
&\boldsymbol{y}|\boldsymbol{\beta}^{\mathrm{R}},\boldsymbol{\beta}^{\mathrm{A}},\boldsymbol{\beta}^{\mathrm{S}},\boldsymbol{u},\sigma^2 \sim \mathrm{N}(\boldsymbol{X}^{\mathrm{R}}\boldsymbol{\beta}^{\mathrm{R}}+\boldsymbol{X}^{\mathrm{A}}\boldsymbol{\beta}^{\mathrm{A}}+\boldsymbol{X}^{\mathrm{S}}\boldsymbol{\beta}^{\mathrm{S}}+\boldsymbol{Z}\boldsymbol{u},\sigma^2\boldsymbol{I}), \quad \boldsymbol{u}|\boldsymbol{G}\sim\mathrm{N}(\boldsymbol{0},\boldsymbol{G}),\\
&\left.\begin{bmatrix}\boldsymbol{\beta}^{\mathrm{R}}\\ \boldsymbol{\beta}^{\mathrm{A}}\\ \boldsymbol{\beta}^{\mathrm{S}}\end{bmatrix}\right|\boldsymbol{\zeta},\tau^2 \sim \mathrm{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}}\\ \boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}}\\ \boldsymbol{0}\end{bmatrix},\begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}} & \boldsymbol{O} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}} & \boldsymbol{O}\\ \boldsymbol{O} & \boldsymbol{O} & \tau^2\mathrm{diag}(\boldsymbol{\zeta})^{-1}\end{bmatrix}\right),\\
&\sigma^2|a_{\sigma^2}\sim\text{Inverse-}\chi^2(\nu_{\sigma^2},1/a_{\sigma^2}), \quad a_{\sigma^2}\sim\text{Inverse-}\chi^2(1,1/(\nu_{\sigma^2}s^2_{\sigma^2})),\\
&\zeta_h|a_{\zeta_h}\overset{\text{ind}}{\sim}\begin{cases}\text{Inverse-}\chi^2(2,1) & \text{for a Laplace prior}\\ \text{Gamma}(1/2,a_{\zeta_h}) & \text{for a Horseshoe prior}\\ \text{Inverse-}\chi^2(2,2a_{\zeta_h}) & \text{for a NEG prior,}\end{cases}\\
&a_{\zeta_h}\overset{\text{ind}}{\sim}\begin{cases}- & \text{for a Laplace prior}\\ \text{Gamma}(1/2,1) & \text{for a Horseshoe prior}\\ \text{Gamma}(\lambda,1) & \text{for a NEG prior,}\end{cases}\\
&\text{for } 1\leq h\leq p_{\mathrm{S}},\\
&\tau^2|a_{\tau^2}\sim\text{Inverse-}\chi^2(1,1/a_{\tau^2}), \quad a_{\tau^2}\sim\text{Inverse-}\chi^2(1,1/s^2_{\tau^2}),\\
&\boldsymbol{G}|\boldsymbol{A}_{\boldsymbol{G}}\sim\mathfrak{p}(\boldsymbol{G}|\boldsymbol{A}_{\boldsymbol{G}}), \quad \boldsymbol{A}_{\boldsymbol{G}}\sim\mathfrak{p}(\boldsymbol{A}_{\boldsymbol{G}}).
\end{aligned}
\tag{3.14}
$$

Here $\boldsymbol{\zeta}\equiv(\zeta_1,\ldots,\zeta_{p_{\mathrm{S}}})^T$ and $\boldsymbol{a}_{\boldsymbol{\zeta}}\equiv(a_{\zeta_1},\ldots,a_{\zeta_{p_{\mathrm{S}}}})^T$. This model can be fitted via MFVB assuming that the full posterior density function is approximated as

$$
\mathfrak{p}(\boldsymbol{\beta},\boldsymbol{u},\sigma^2,a_{\sigma^2},\boldsymbol{\zeta},\boldsymbol{a}_{\boldsymbol{\zeta}},\tau^2,a_{\tau^2},\boldsymbol{G},\boldsymbol{A}_{\boldsymbol{G}}|\boldsymbol{y})\approx\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u},\sigma^2,a_{\sigma^2},\boldsymbol{\zeta},\boldsymbol{a}_{\boldsymbol{\zeta}},\tau^2,a_{\tau^2},\boldsymbol{G},\boldsymbol{A}_{\boldsymbol{G}}) \tag{3.15}
$$

and a tractable solution arises with the following mean-field restriction:

$$
\begin{aligned}
&\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u},\sigma^2,a_{\sigma^2},\boldsymbol{\zeta},\boldsymbol{a}_{\boldsymbol{\zeta}},\tau^2,a_{\tau^2},\boldsymbol{G},\boldsymbol{A}_{\boldsymbol{G}})=\\
&\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})\,\mathfrak{q}(\sigma^2)\,\mathfrak{q}(a_{\sigma^2})\left\{\prod_{h=1}^{p_{\mathrm{S}}}\mathfrak{q}(\zeta_h)\,\mathfrak{q}(a_{\zeta_h})\right\}\mathfrak{q}(\tau^2)\,\mathfrak{q}(a_{\tau^2})\,\mathfrak{q}(\boldsymbol{G})\,\mathfrak{q}(\boldsymbol{A}_{\boldsymbol{G}}).
\end{aligned}
\tag{3.16}
$$

Arguments similar to those given in Sections 3.3.1 and 3.4.2 lead to the optimal approximating densities being:

$$
\begin{gathered}
\mathfrak{q}^*(\boldsymbol{\beta}, \boldsymbol{u}) \text{ is a } \mathrm{N}\big(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}\big) \text{ density function,} \\
\mathfrak{q}^*(\sigma^2) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(\sigma^2)}, \lambda_{\mathfrak{q}(\sigma^2)}\big) \text{ density function,} \\
\mathfrak{q}^*(a_{\sigma^2}) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(a_{\sigma^2})}, \lambda_{\mathfrak{q}(a_{\sigma^2})}\big) \text{ density function,} \\
\mathfrak{q}^*(\zeta_h) \text{ is } \begin{cases} \text{an Inverse-Gaussian}(\mu_{\mathfrak{q}(\zeta_h)}, 1) \text{ density function} & \text{for a Laplace prior} \\ \text{a Gamma}(1, \lambda_{\mathfrak{q}(\zeta_h)}) \text{ density function} & \text{for a Horseshoe prior} \\ \text{an Inverse-Gaussian}(\mu_{\mathfrak{q}(\zeta_h)}, \lambda_{\mathfrak{q}(\zeta_h)}) \text{ density function} & \text{for a NEG prior,} \end{cases} \\
\mathfrak{q}^*(a_{\zeta_h}) \text{ is } \begin{cases} - & \text{for a Laplace prior} \\ \text{a Gamma}(1, \lambda_{\mathfrak{q}(a_{\zeta_h})}) \text{ density function} & \text{for a Horseshoe prior} \\ \text{a Gamma}(\lambda + 1, \lambda_{\mathfrak{q}(a_{\zeta_h})}) \text{ density function} & \text{for a NEG prior,} \end{cases} \\
\text{for } 1 \leq h \leq p_{\mathrm{S}}, \\
\mathfrak{q}^*(\tau^2) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(\tau^2)}, \lambda_{\mathfrak{q}(\tau^2)}\big) \text{ density function} \\
\text{and } \mathfrak{q}^*(a_{\tau^2}) \text{ is an Inverse-}\chi^2\big(\xi_{\mathfrak{q}(a_{\tau^2})}, \lambda_{\mathfrak{q}(a_{\tau^2})}\big) \text{ density function.}
\end{gathered}
\tag{3.17}
$$

The optimal approximating densities for the matrices $\boldsymbol{G}$ and $\boldsymbol{A}_G$ vary according to the adopted random effect structure, as explained in Section 3.3. Notice that (3.16) jointly approximates $\boldsymbol{\beta}^{\mathrm{R}}$, $\boldsymbol{\beta}^{\mathrm{A}}$, $\boldsymbol{\beta}^{\mathrm{S}}$ and $\boldsymbol{u}$, allowing all the fixed effects to share posterior dependence with the random effects. Also, all the $\mathfrak{q}^*(\beta_h^{\mathrm{S}})$ approximating densities are Gaussian, although different global-local prior specifications may lead to marginal posterior density functions $\mathfrak{p}(\beta_h^{\mathrm{S}}|\boldsymbol{y})$ having shapes around zero that are different from the typical bell-shaped behavior, especially those of fixed effects associated to irrelevant covariates. Using MFVB, we sacrifice some degree of accuracy yet obtaining a substantial computational time advantage over standard MCMC sampling procedures.

3.5.1 Naïve MFVB Updates

The updates of the parameters of the $\mathfrak{q}$-densities in (3.17) can be derived combining and adapting the results discussed in Sections 3.3.1 and 3.4.2. Those related to

the $\mathfrak{q}^*(\boldsymbol{\beta}, \boldsymbol{u})$ optimal density function are:

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} &\longleftarrow \left(\mu_{\mathfrak{q}(1/\sigma^2)}\, \boldsymbol{C}^T\boldsymbol{C} + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}}^{-1} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}}^{-1} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \mu_{\mathfrak{q}(1/\tau^2)}\,\mathrm{diag}(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\zeta})}) & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & E_{\mathfrak{q}}(\boldsymbol{G}^{-1}) \end{bmatrix} \right)^{-1} \\
\text{and} \quad \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} &\longleftarrow \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} \left(\mu_{\mathfrak{q}(1/\sigma^2)}\, \boldsymbol{C}^T\boldsymbol{y} + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}} \\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \right).
\end{aligned} \tag{3.18}$$

The update for $E_{\mathfrak{q}}(\boldsymbol{G}^{-1})$ is given by (3.8) or (3.9) depending on whether a two-level or a three-level random effects specification is considered. The main difference with the expressions given in (3.7) is that a global-local prior specification on each element of $\boldsymbol{\beta}^{\mathrm{S}}$ introduces the diagonal matrix $\mu_{\mathfrak{q}(1/\tau^2)}\,\mathrm{diag}(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\zeta})})$ of dimension $p_{\mathrm{S}} \times p_{\mathrm{S}}$ inside the update expression for $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$. This matrix is updated at each iteration of the MFVB algorithm employing the updated values of $\mu_{\mathfrak{q}(1/\tau^2)}$ and $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\zeta})}$, accordingly to the global-local prior adopted.

The updates for the parameters of $\mathfrak{q}^*(\sigma^2)$ and $\mathfrak{q}^*(a_{\sigma^2})$ are identical to those in (3.7). The updates for the parameters of the optimal approximating $\mathfrak{q}$-densities of $\boldsymbol{G}$ and $\boldsymbol{A}_G$ are identical to those described in Section 3.3.1. The updates for the parameters of $\mathfrak{q}^*(\zeta_h)$ and $\mathfrak{q}^*(a_{\zeta_h})$, $1 \leq h \leq p_{\mathrm{S}}$, $\mathfrak{q}^*(\tau^2)$ and $\mathfrak{q}^*(a_{\tau^2})$ follow with minor modifications from those mentioned in Section 3.4.2, replacing H with p_{S}.

Nevertheless, naïve updates (3.18) suffer from the same problems elucidated in Section 3.3 for large m values. Therefore, an appropriate streamlined enhancement for efficient implementations is required.

3.5.2 Streamlined MFVB Updates

Results 1 and 4 from Nolan *et al.* (2020) can be extended to derive a streamlined MFVB algorithm for efficiently updating the parameters of the densities in (3.17) and, in particular, for exploiting the sparse matrix structures presented in updates (3.18). Such results are based on the two- and three-level *sparse matrix least squares problems* defined in Nolan and Wand (2020) and make use of the associated SolveTwoLevelSparseMatrix and SolveThreeLevelSparseMatrix routines, which are recalled in Appendix C.1.

The following results explain how to efficiently compute $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ and the relevant sub-blocks of $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ which are necessary for finding the optimal $\mathfrak{q}$-densities in (3.17).

Result 3.1. *The MFVB updates of* $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ *and each of the sub-blocks of* $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ *that are relevant for variational inference concerning model* (3.14) *with a two-level random effects specification are expressible as a two-level sparse matrix problem of the form* $\boldsymbol{A}\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} = \boldsymbol{a}$*, where* $\boldsymbol{a}$ *and the non-zero sub-blocks of* $\boldsymbol{A}$ *are, according to the notation in Appendix C.1:*

$$\boldsymbol{A}_{11} = \mu_{\mathfrak{q}(1/\sigma^2)}\sum_{i=1}^{m}(\boldsymbol{X}_i^T\boldsymbol{X}_i) + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}^R}^{-1} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Sigma}_{\boldsymbol{\beta}^A}^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \mu_{\mathfrak{q}(1/\tau^2)}\, diag(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\zeta})}) \end{bmatrix},$$

$$\boldsymbol{a}_1 = \mu_{\mathfrak{q}(1/\sigma^2)}\sum_{i=1}^{m}(\boldsymbol{X}_i^T\boldsymbol{y}_i) + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}^R}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}^R} \\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}^A}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}^A} \\ \boldsymbol{0} \end{bmatrix},$$

$$\boldsymbol{A}_{22,i} = \mu_{\mathfrak{q}(1/\sigma^2)}\boldsymbol{Z}_i^T\boldsymbol{Z}_i + \boldsymbol{M}_{q(\boldsymbol{\Sigma}^{-1})}, \quad \boldsymbol{A}_{12,i} = \mu_{\mathfrak{q}(1/\sigma^2)}\boldsymbol{X}_i^T\boldsymbol{Z}_i, \quad \boldsymbol{a}_{2,i} = \mu_{\mathfrak{q}(1/\sigma^2)}\boldsymbol{Z}_i^T\boldsymbol{y}_i,$$

for $1 \leq i \leq m$*. Moreover,* $\boldsymbol{X}_i = [\, \boldsymbol{X}_i^R \mid \boldsymbol{X}_i^A \mid \boldsymbol{X}_i^S \,]$.

The SolveTwoLevelSparseMatrix *routine efficiently solves the associated linear system and provides the solutions:*

$$\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})} = \boldsymbol{x}_1, \quad \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta})} = \boldsymbol{A}^{11}$$

and

$$\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i)} = \boldsymbol{x}_{2,i}, \quad \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_i)} = \boldsymbol{A}^{22,i}, \quad E_{\mathfrak{q}}\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})})(\boldsymbol{u}_i - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i)})^T\} = \boldsymbol{A}^{12,i}, \quad 1 \leq i \leq m.$$

Result 3.2. *The MFVB updates of* $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ *and each of the sub-blocks of* $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})}$ *that are relevant for variational inference concerning model* (3.14) *with a three-level random effects specification are expressible as a three-level sparse matrix problem of the form* $\boldsymbol{A}\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta},\boldsymbol{u})} = \boldsymbol{a}$*,*

where $\boldsymbol{a}$ *and the non-zero sub-blocks of* $\boldsymbol{A}$ *are, according to the notation in Appendix C.1:*

$$\boldsymbol{A}_{11} = \mu_{\mathfrak{q}(1/\sigma^2)} \sum_{i=1}^{m} \sum_{j=1}^{n_i} (\boldsymbol{X}_{ij}^T \boldsymbol{X}_{ij}) + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}^R}^{-1} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Sigma}_{\boldsymbol{\beta}^A}^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \mu_{\mathfrak{q}(1/\tau^2)}\, diag(\boldsymbol{\mu}_{\mathfrak{q}(\zeta)}) \end{bmatrix},$$

$$\boldsymbol{a}_1 = \mu_{\mathfrak{q}(1/\sigma^2)} \sum_{i=1}^{m} \sum_{j=1}^{n_i} (\boldsymbol{X}_{ij}^T \boldsymbol{y}_{ij}) + \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}^R}^{-1} \boldsymbol{\mu}_{\boldsymbol{\beta}^R} \\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}^A}^{-1} \boldsymbol{\mu}_{\boldsymbol{\beta}^A} \\ \boldsymbol{0} \end{bmatrix},$$

$$\boldsymbol{A}_{22,i} = \mu_{\mathfrak{q}(1/\sigma^2)} \sum_{j=1}^{n_i} ((\boldsymbol{Z}_{ij}^{L1})^T \boldsymbol{Z}_{ij}^{L1}) + \boldsymbol{M}_{q((\boldsymbol{\Sigma}^{L1})^{-1})}, \quad \boldsymbol{A}_{12,i} = \mu_{\mathfrak{q}(1/\sigma^2)} \sum_{j=1}^{n_i} (\boldsymbol{X}_{ij})^T \boldsymbol{Z}_{ij}^{L1},$$

$$\boldsymbol{a}_{2,i} = \mu_{\mathfrak{q}(1/\sigma^2)} \sum_{j=1}^{n_i} (\boldsymbol{Z}_{ij}^{L1})^T \boldsymbol{y}_{ij}, \quad \boldsymbol{A}_{22,ij} = \mu_{\mathfrak{q}(1/\sigma^2)} (\boldsymbol{Z}_{ij}^{L2})^T \boldsymbol{Z}_{ij}^{L2} + \boldsymbol{M}_{q((\boldsymbol{\Sigma}^{L2})^{-1})},$$

$$\boldsymbol{A}_{12,i,j} = \mu_{\mathfrak{q}(1/\sigma^2)} (\boldsymbol{Z}_{ij}^{L1})^T \boldsymbol{Z}_{ij}^{L2}, \quad \boldsymbol{A}_{12,ij} = \mu_{\mathfrak{q}(1/\sigma^2)} \boldsymbol{X}_{ij}^T \boldsymbol{Z}_{ij}^{L2}, \quad \boldsymbol{a}_{2,ij} = \mu_{\mathfrak{q}(1/\sigma^2)} (\boldsymbol{Z}_{ij}^{L2})^T \boldsymbol{y}_{ij},$$

for $1 \leq i \leq m$ *and* $1 \leq j \leq n_i$. *Moreover,* $\boldsymbol{X}_{ij} = [\, \boldsymbol{X}_{ij}^R \mid \boldsymbol{X}_{ij}^A \mid \boldsymbol{X}_{ij}^S \,]$.

The SolveThreeLevelSparseMatrix *routine efficiently solves the associated linear system, and provides the solutions:*

$$\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})} = \boldsymbol{x}_1, \quad \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\beta})} = \boldsymbol{A}^{11},$$

$$\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i^{L1})} = \boldsymbol{x}_{2,i}, \quad \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_i^{L1})} = \boldsymbol{A}^{22,i}, \quad E_{\mathfrak{q}}\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})})(\boldsymbol{u}_i^{L1} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i^{L1})})^T\} = \boldsymbol{A}^{12,i}, \quad 1 \leq i \leq m,$$

and

$$\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{L2})} = \boldsymbol{x}_{2,ij}, \quad \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{L2})} = \boldsymbol{A}^{22,ij}, \quad E_{\mathfrak{q}}\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\beta})})(\boldsymbol{u}_{ij}^{L2} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{L2})}^T\} = \boldsymbol{A}^{12,ij},$$

$$E_{\mathfrak{q}}\{(\boldsymbol{u}_i^{L1} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_i^{L1})})(\boldsymbol{u}_{ij}^{L2} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{u}_{ij}^{L2})})^T\} = \boldsymbol{A}^{12,i,j}, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n_i.$$

We employ these two results to derive a streamlined MFVB algorithm for determining the optimal parameters of the densities in (3.17) for multilevel models having two-level and three-level random effects. The former specification is accommodated by Algorithm 3.2, the latter by Algorithm 3.3. Notice all the sub-blocks of $\boldsymbol{A}$ and components of $\boldsymbol{a}$ described in Results 3.1 and 3.2 can be updated performing linear transformations of matrices involving multiplications of sub-vectors of $\boldsymbol{y}$ and sub-matrices of $\boldsymbol{X}$ and $\boldsymbol{Z}$, which need to be computed only once instead of at each iteration of the associated algorithms.

These results are inspired by Results 1 and 4 of Nolan *et al.* (2020), which are

expressed in terms of *sparse least-squares problems* of the type $\|\boldsymbol{b} - \boldsymbol{B}\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})}\|^2$, after exploiting the equalities $\boldsymbol{A} = \boldsymbol{B}^T\boldsymbol{B}$ and $\boldsymbol{a} = \boldsymbol{B}^T\boldsymbol{b}$. This class of problems can be solved via efficient QR-decompositions, instead of sparse matrix problems of the type $\boldsymbol{A}\boldsymbol{\mu}_{q(\boldsymbol{\beta},\boldsymbol{u})} = \boldsymbol{a}$ which rely upon matrix inversion routines. Section 2.1 of Nolan and Wand (2020) claims that the former class of problems and associated SolveTwoLevelSparseLeastSquares and SolveThreeLevelSparseLeastSquares routines proposed in Appendix A of Nolan *et al.* (2020) are numerically preferred to the latter, since QR-decomposition methods are more computationally stable. However, streamlined MFVB approximations based on the QR-decomposition require additional matrices to be updated at each algorithm iteration. Efficient QR-decomposition routines and functions for performing matrix multiplications with the associated $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are unavailable in all standard computing environments and programming languages. Furthermore, the sub-blocks of $\boldsymbol{B}$ and components of $\boldsymbol{b}$ become particularly sparse for large p values, as for the cases considered in this work, entailing onerous memory consumption and compromising the efficiency of QR-decomposition. For these reasons, we opt for routines based on sparse matrix problems instead of those based on sparse least-squares problems.

The convergence of Algorithms 3.2 and 3.3 can be assessed in several ways. One way is to monitor the relative increment of the variational lower bound and stop it whenever it falls below a pre-specified threshold. The drawback is that much tedious algebra is required to derive an explicit lower bound expression. An alternative way is to measure the absolute relative increments among all the parameters updated at each iteration and stop when the maximum increment falls below a pre-specified threshold. However, both the approaches can be computationally expensive, especially for large m values, and these convergence checks may significantly slow down the overall streamlined iterations. Therefore, we suggest letting Algorithms 3.2 and 3.3 run for a reasonable number of iterations fixed in advance and increase the number of iterations if convergence checks suggest that the desired level of convergence has not been achieved. Although any formal theoretical guarantees of convergence are provided, this technique embeds faster iterations, which remains the main benefit of streamlined variational techniques.

Algorithm 3.2 *Streamlined algorithm for obtaining the mean field variational Bayes approximate posterior density functions* (3.17) *for the parameters of the linear mixed model* (3.14) *with the two-level random effects specification. The approximation is based on the mean-field density restriction* (3.16)*. The algorithm description requires more than one page and is continued on a subsequent page.*

Data Inputs: $\boldsymbol{y}_i\ (o_i \times 1), \boldsymbol{X}_i^{\mathrm{R}}\ (o_i \times p_{\mathrm{R}}), \boldsymbol{X}_i^{\mathrm{A}}\ (o_i \times p_{\mathrm{A}}), \boldsymbol{X}_i^{\mathrm{S}}\ (o_i \times p_{\mathrm{S}}), \boldsymbol{Z}_i\ (o_i \times q),\ 1 \leq i \leq m.$

Build $\boldsymbol{X}_i = \left[\, \boldsymbol{X}_i^{\mathrm{R}} \,\middle|\, \boldsymbol{X}_i^{\mathrm{A}} \,\middle|\, \boldsymbol{X}_i^{\mathrm{S}} \,\right]\ (o_i \times p).$

Global-local prior type choice: Laplace, Horseshoe or NEG.

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}}\ (p_{\mathrm{R}} \times 1), \boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}}\ (p_{\mathrm{A}} \times 1), \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}}\ (p_{\mathrm{R}} \times p_{\mathrm{R}})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}}\ (p_{\mathrm{A}} \times p_{\mathrm{A}})$ both symmetric and positive definite, $\nu_{\sigma^2}, s_{\sigma^2}, s_{\tau^2}, \nu_{\boldsymbol{\Sigma}}, s_{\boldsymbol{\Sigma},1}, \ldots, s_{\boldsymbol{\Sigma},q} > 0.$

If NEG: $\lambda > 0.$

Initialize: $\mu_{\mathrm{q}(1/\sigma^2)} > 0, \mu_{\mathrm{q}(1/a_{\sigma^2})} > 0, \mu_{\mathrm{q}(1/\tau^2)} > 0, \mu_{\mathrm{q}(1/a_{\tau^2})} > 0, \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\zeta})}\ (p_{\mathrm{S}} \times 1), \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{a}_{\boldsymbol{\zeta}})}\ (p_{\mathrm{S}} \times 1),$
$\boldsymbol{M}_{\mathrm{q}(\boldsymbol{\Sigma}^{-1})}\ (q \times q), \boldsymbol{M}_{\mathrm{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}}^{-1})}\ (q \times q)$ both symmetric and positive definite.

$\xi_{\mathrm{q}(\sigma^2)} \longleftarrow \nu_{\sigma^2} + \sum_{i=1}^{m} o_i; \quad \xi_{\mathrm{q}(\boldsymbol{\Sigma})} \longleftarrow \nu_{\boldsymbol{\Sigma}} + m + 2q - 2; \quad \xi_{\mathrm{q}(a_{\sigma^2})} \longleftarrow \nu_{\sigma^2} + 1; \quad \xi_{\mathrm{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}})} \longleftarrow \nu_{\boldsymbol{\Sigma}} + q;$

$\xi_{\mathrm{q}(\tau^2)} \longleftarrow p_{\mathrm{S}} + 1; \quad \xi_{\mathrm{q}(a_{\tau^2})} \longleftarrow 2; \quad \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}}} \longleftarrow \{\nu_{\boldsymbol{\Sigma}}\, \mathrm{diag}(s_{\boldsymbol{\Sigma},1}^2, \ldots, s_{\boldsymbol{\Sigma},q}^2)\}^{-1}.$

Cycle until convergence:

Compute $\boldsymbol{a}_1, \boldsymbol{A}_{11}, \{\boldsymbol{a}_{2,i}, \boldsymbol{A}_{22,i}, \boldsymbol{A}_{12,i} : 1 \leq i \leq m\}$ with expressions from Result 3.1.

$\mathcal{S}_1 \longleftarrow \textsc{SolveTwoLevelSparseMatrix}\Big(\boldsymbol{a}_1, \boldsymbol{A}_{11}, \{\boldsymbol{a}_{2,i}, \boldsymbol{A}_{22,i}, \boldsymbol{A}_{12,i} : 1 \leq i \leq m\}\Big).$

$\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})} \longleftarrow \boldsymbol{x}_1$ component of $\mathcal{S}_1; \quad \boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\beta})} \longleftarrow \boldsymbol{A}^{11}$ component of $\mathcal{S}_1;$

$\boldsymbol{\mu}_{\mathrm{q}((\boldsymbol{\beta}^{\mathrm{S}})^2)} \longleftarrow \mathrm{diagonal}\left(\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\beta}^{\mathrm{S}})} + \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta}^{\mathrm{S}})}\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta}^{\mathrm{S}})}^T\right);$

$\lambda_{\mathrm{q}(\sigma^2)} \longleftarrow \mu_{\mathrm{q}(1/a_{\sigma^2})}; \quad \boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma})} \longleftarrow \boldsymbol{M}_{\mathrm{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}}^{-1})}.$

For $i = 1, \ldots, m$:

$\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}_i)} \longleftarrow \boldsymbol{x}_{2,i}$ component of $\mathcal{S}_1; \quad \boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{u}_i)} \longleftarrow \boldsymbol{A}^{22,i}$ component of $\mathcal{S}_1;$

$E_{\mathrm{q}}\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})})(\boldsymbol{u}_i - \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}_i)})^T\} \longleftarrow \boldsymbol{A}^{12,i}$ component of $\mathcal{S}_1;$

$\lambda_{\mathrm{q}(\sigma^2)} \longleftarrow \lambda_{\mathrm{q}(\sigma^2)} + \left\|\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})} - \boldsymbol{Z}_i\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}_i)}\right\|^2 + \mathrm{tr}(\boldsymbol{X}_i^T\boldsymbol{X}_i\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\beta})}) + \mathrm{tr}(\boldsymbol{Z}_i^T\boldsymbol{Z}_i\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{u}_i)})$
$+ 2\mathrm{tr}[\boldsymbol{Z}_i^T\boldsymbol{X}_i E_{\mathrm{q}}\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})})(\boldsymbol{u}_i - \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}_i)})^T\}];$

$\boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma})} \longleftarrow \boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma})} + \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}_i)}\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}_i)}^T + \boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{u}_i)};$

$\mu_{\mathrm{q}(1/\sigma^2)} \longleftarrow \xi_{\mathrm{q}(\sigma^2)}/\lambda_{\mathrm{q}(\sigma^2)}; \quad \boldsymbol{M}_{\mathrm{q}(\boldsymbol{\Sigma}^{-1})} \longleftarrow (\xi_{\mathrm{q}(\boldsymbol{\Sigma})} - q + 1)\boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma})}^{-1};$

$\lambda_{\mathrm{q}(a_{\sigma^2})} \longleftarrow \mu_{\mathrm{q}(1/\sigma^2)} + 1/(\nu_{\sigma^2}s_{\sigma^2}^2); \quad \mu_{\mathrm{q}(1/a_{\sigma^2})} \longleftarrow \xi_{\mathrm{q}(a_{\sigma^2})}/\lambda_{\mathrm{q}(a_{\sigma^2})};$

$\lambda_{\mathrm{q}(\tau^2)} \longleftarrow \mu_{\mathrm{q}(1/a_{\tau^2})} + \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\zeta})}^T\boldsymbol{\mu}_{\mathrm{q}((\boldsymbol{\beta}^{\mathrm{S}})^2)}; \quad \mu_{\mathrm{q}(1/\tau^2)} \longleftarrow \xi_{\mathrm{q}(\tau^2)}/\lambda_{\mathrm{q}(\tau^2)};$

$\lambda_{\mathrm{q}(a_{\tau^2})} \longleftarrow \mu_{\mathrm{q}(1/\tau^2)} + 1/s_{\tau^2}^2; \quad \mu_{\mathrm{q}(1/a_{\tau^2})} \longleftarrow \xi_{\mathrm{q}(a_{\tau^2})}/\lambda_{\mathrm{q}(a_{\tau^2})};$

$\boldsymbol{g} \longleftarrow \frac{1}{2}\mu_{\mathrm{q}(1/\tau^2)}\boldsymbol{\mu}_{\mathrm{q}((\boldsymbol{\beta}^{\mathrm{S}})^2)};$

continued on a subsequent page ...

Algorithm 3.2 continued. *This is a continuation of the description of this algorithm that commences on a preceding page.*

If Laplace: $\boldsymbol{\mu}_{q(\zeta)} \longleftarrow \sqrt{\mathbf{1}/(2\boldsymbol{g})}$;

If Horseshoe: $\boldsymbol{\lambda}_{q(\zeta)} \longleftarrow \boldsymbol{\mu}_{q(a_\zeta)} + \boldsymbol{g}$; $\quad \boldsymbol{\mu}_{q(\zeta)} \longleftarrow \mathbf{1}/\boldsymbol{\lambda}_{q(\zeta)}$;

$\boldsymbol{\lambda}_{q(a_\zeta)} \longleftarrow \boldsymbol{\mu}_{q(\zeta)} + \mathbf{1}$; $\quad \boldsymbol{\mu}_{q(a_\zeta)} \longleftarrow \mathbf{1}/\boldsymbol{\lambda}_{q(a_\zeta)}$;

If NEG: $\boldsymbol{\lambda}_{q(\zeta)} \longleftarrow 2\boldsymbol{\mu}_{q(a_\zeta)}$; $\quad \boldsymbol{\mu}_{q(\zeta)} \longleftarrow \sqrt{\boldsymbol{\lambda}_{q(\zeta)}/(2\boldsymbol{g})}$; $\quad \boldsymbol{\mu}_{q(\mathbf{1}/\zeta)} \longleftarrow \mathbf{1}/\boldsymbol{\mu}_{q(\zeta)} + \mathbf{1}/(2\boldsymbol{\mu}_{q(a_\zeta)})$;

$\boldsymbol{\lambda}_{q(a_\zeta)} \longleftarrow \boldsymbol{\mu}_{q(\mathbf{1}/\zeta)} + \mathbf{1}$; $\quad \boldsymbol{\mu}_{q(a_\zeta)} \longleftarrow (\lambda+1)(\mathbf{1}/\boldsymbol{\lambda}_{q(a_\zeta)})$;

$\boldsymbol{\Lambda}_{q(\boldsymbol{A}_\Sigma)} \longleftarrow \text{diag}\{\text{diagonal}(\boldsymbol{M}_{q(\boldsymbol{\Sigma}^{-1})})\} + \boldsymbol{\Lambda}_{\boldsymbol{A}_\Sigma}$; $\quad \boldsymbol{M}_{q(\boldsymbol{A}_\Sigma^{-1})} \longleftarrow \xi_{q(\boldsymbol{A}_\Sigma)}\boldsymbol{\Lambda}_{q(\boldsymbol{A}_\Sigma)}^{-1}$.

Outputs: $\boldsymbol{\mu}_{q(\boldsymbol{\beta})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}, \left\{\boldsymbol{\mu}_{q(\boldsymbol{u}_i)}, \boldsymbol{\Sigma}_{q(\boldsymbol{u}_i)}, E_q\{(\boldsymbol{\beta}-\boldsymbol{\mu}_{q(\boldsymbol{\beta})})(\boldsymbol{u}_i-\boldsymbol{\mu}_{q(\boldsymbol{u}_i)})^T\} : 1 \le i \le m\right\}$,

$\xi_{q(\sigma^2)}, \lambda_{q(\sigma^2)}, \xi_{q(a_{\sigma^2})}, \lambda_{q(a_{\sigma^2})}, \xi_{q(\tau^2)}, \lambda_{q(\tau^2)}, \xi_{q(a_{\tau^2})}, \lambda_{q(a_{\tau^2})}$,

if Laplace: $\boldsymbol{\mu}_{q(\zeta)}$, if Horseshoe: $\left\{\boldsymbol{\lambda}_{q(\zeta)}, \boldsymbol{\lambda}_{q(a_\zeta)}\right\}$, if NEG: $\left\{\boldsymbol{\mu}_{q(\zeta)}, \boldsymbol{\lambda}_{q(\zeta)}, \boldsymbol{\lambda}_{q(a_\zeta)}\right\}$,

$\xi_{q(\boldsymbol{\Sigma})}, \boldsymbol{\Lambda}_{q(\boldsymbol{\Sigma})}, \xi_{q(\boldsymbol{A}_\Sigma)}, \boldsymbol{\Lambda}_{q(\boldsymbol{A}_\Sigma)}$.

Algorithm 3.3 *Streamlined algorithm for obtaining the mean field variational Bayes approximate posterior density functions* (3.17) *for the parameters of the linear mixed model* (3.14) *with the three-level random effects specification. The approximation is based on the mean-field density restriction* (3.16)*. The algorithm description requires more than one page and is continued on a subsequent page.*

Data Inputs: $\boldsymbol{y}_{ij}\ (o_{ij}\times 1), \boldsymbol{X}^{\mathrm{R}}_{ij}\ (o_{ij}\times p_{\mathrm{R}}), \boldsymbol{X}^{\mathrm{A}}_{ij}\ (o_{ij}\times p_{\mathrm{A}}), \boldsymbol{X}^{\mathrm{S}}_{ij}\ (o_{ij}\times p_{\mathrm{S}}),$
$\boldsymbol{Z}^{\mathrm{L1}}_{ij}\ (o_{ij}\times q_1), \boldsymbol{Z}^{\mathrm{L2}}_{ij}\ (o_{ij}\times q_2),\ 1\le i\le m,\ 1\le j\le n_i.$
Build $\boldsymbol{X}_{ij}=\left[\boldsymbol{X}^{\mathrm{R}}_{ij}\,\middle|\,\boldsymbol{X}^{\mathrm{A}}_{ij}\,\middle|\,\boldsymbol{X}^{\mathrm{S}}_{ij}\right]\ (n_i\times p).$

Global-local prior type choice: Laplace, Horseshoe or NEG.

Hyperparameter Inputs: $\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}}\ (p_{\mathrm{R}}\times 1), \boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}}\ (p_{\mathrm{A}}\times 1), \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}}\ (p_{\mathrm{R}}\times p_{\mathrm{R}})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}}\ (p_{\mathrm{A}}\times p_{\mathrm{A}})$ both symmetric and positive definite, $\nu_{\sigma^2}, s_{\sigma^2}, s_{\tau^2}, \nu_{\boldsymbol{\Sigma}^{\mathrm{L1}}}, \nu_{\boldsymbol{\Sigma}^{\mathrm{L2}}}$, $s_{\boldsymbol{\Sigma}^{\mathrm{L1}},1},\ldots,s_{\boldsymbol{\Sigma}^{\mathrm{L1}},q_1}, s_{\boldsymbol{\Sigma}^{\mathrm{L2}},1},\ldots,s_{\boldsymbol{\Sigma}^{\mathrm{L2}},q_2}>0$. If NEG: $\lambda>0$.

Initialize: $\mu_{\mathrm{q}(1/\sigma^2)}>0, \mu_{\mathrm{q}(1/a_{\sigma^2})}>0, \mu_{\mathrm{q}(1/\tau^2)}>0, \mu_{\mathrm{q}(1/a_{\tau^2})}>0, \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\zeta})}\ (p_{\mathrm{S}}\times 1), \boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{a}_{\boldsymbol{\zeta}})}\ (p_{\mathrm{S}}\times 1),$
$\boldsymbol{M}_{\mathrm{q}((\boldsymbol{\Sigma}^{\mathrm{L1}})^{-1})}\ (q_1\times q_1), \boldsymbol{M}_{\mathrm{q}((\boldsymbol{\Sigma}^{\mathrm{L2}})^{-1})}\ (q_2\times q_2), \boldsymbol{M}_{\mathrm{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\mathrm{L1}}})}\ (q_1\times q_1), \boldsymbol{M}_{\mathrm{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\mathrm{L2}}})}\ (q_2\times q_2)$
all symmetric and positive definite.

$\xi_{\mathrm{q}(\sigma^2)}\longleftarrow \nu_{\sigma^2}+\sum_{i=1}^{m}\sum_{j=1}^{n_i}o_{ij};\quad \xi_{\mathrm{q}(\boldsymbol{\Sigma}^{\mathrm{L1}})}\longleftarrow \nu_{\boldsymbol{\Sigma}^{\mathrm{L1}}}+m+2q_1-2;\quad \xi_{\mathrm{q}(\boldsymbol{\Sigma}^{\mathrm{L2}})}\longleftarrow \nu_{\boldsymbol{\Sigma}^{\mathrm{L2}}}+\sum_{i=1}^{m}n_i+2q_2-2;$

$\xi_{\mathrm{q}(a_{\sigma^2})}\longleftarrow \nu_{\sigma^2}+1;\quad \xi_{\mathrm{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\mathrm{L1}}})}\longleftarrow \nu_{\boldsymbol{\Sigma}^{\mathrm{L1}}}+q_1;\quad \xi_{\mathrm{q}(\boldsymbol{A}_{\boldsymbol{\Sigma}^{\mathrm{L2}}})}\longleftarrow \nu_{\boldsymbol{\Sigma}^{\mathrm{L2}}}+q_2;$

$\xi_{\mathrm{q}(\tau^2)}\longleftarrow p_{\mathrm{S}}+1;\quad \xi_{\mathrm{q}(a_{\tau^2})}\longleftarrow 2;$

$\boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\mathrm{L1}}}}\longleftarrow \{\nu_{\boldsymbol{\Sigma}^{\mathrm{L1}}}\,\mathrm{diag}(s^2_{\boldsymbol{\Sigma}^{\mathrm{L1}},1},\ldots,s^2_{\boldsymbol{\Sigma}^{\mathrm{L1}},q_1})\}^{-1};\quad \boldsymbol{\Lambda}_{\boldsymbol{A}_{\boldsymbol{\Sigma}^{\mathrm{L2}}}}\longleftarrow \{\nu_{\boldsymbol{\Sigma}^{\mathrm{L2}}}\,\mathrm{diag}(s^2_{\boldsymbol{\Sigma}^{\mathrm{L2}},1},\ldots,s^2_{\boldsymbol{\Sigma}^{\mathrm{L2}},q_2})\}^{-1}.$

Cycle until convergence:

Compute $\boldsymbol{a}_1, \boldsymbol{A}_{11}, \{\boldsymbol{a}_{2,i}, \boldsymbol{A}_{22,i}, \boldsymbol{A}_{12,i} : 1\le i\le m\}, \{\boldsymbol{a}_{2,ij}, \boldsymbol{A}_{22,ij}, \boldsymbol{A}_{12,ij}, \boldsymbol{A}_{12,i,j} :$
$1\le i\le m, 1\le j\le n_i\}$ with expressions from Result 3.2.

$\mathcal{S}_2\longleftarrow \textsc{SolveThreeLevelSparseMatrix}\Big(\boldsymbol{a}_1, \boldsymbol{A}_{11}, \{\boldsymbol{a}_{2,i}, \boldsymbol{A}_{22,i}, \boldsymbol{A}_{12,i} : 1\le i\le m\},$
$\{\boldsymbol{a}_{2,ij}, \boldsymbol{A}_{22,ij}, \boldsymbol{A}_{12,ij}, \boldsymbol{A}_{12,i,j} : 1\le i\le m, 1\le j\le n_i\}\Big).$

$\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})}\longleftarrow \boldsymbol{x}_1$ component of $\mathcal{S}_2;\quad \boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\beta})}\longleftarrow \boldsymbol{A}^{11}$ component of $\mathcal{S}_2;$

$\boldsymbol{\mu}_{\mathrm{q}((\boldsymbol{\beta}^{\mathrm{S}})^2)}\longleftarrow \mathrm{diagonal}\left(\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\beta}^{\mathrm{S}})}+\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta}^{\mathrm{S}})}\boldsymbol{\mu}^T_{\mathrm{q}(\boldsymbol{\beta}^{\mathrm{S}})}\right);$

$\lambda_{\mathrm{q}(\sigma^2)}\longleftarrow \mu_{\mathrm{q}(1/a_{\sigma^2})};\quad \boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma}^{\mathrm{L1}})}\longleftarrow \boldsymbol{M}_{\mathrm{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\mathrm{L1}}})};\quad \boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma}^{\mathrm{L2}})}\longleftarrow \boldsymbol{M}_{\mathrm{q}(\boldsymbol{A}^{-1}_{\boldsymbol{\Sigma}^{\mathrm{L2}}})}.$

For $i=1,\ldots,m$:

$\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)}\longleftarrow \boldsymbol{x}_{2,i}$ component of $\mathcal{S}_2;\quad \boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)}\longleftarrow \boldsymbol{A}^{22,i}$ component of $\mathcal{S}_2;$
$E_{\mathrm{q}}\{(\boldsymbol{\beta}-\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})})(\boldsymbol{u}^{\mathrm{L1}}_i-\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)})^T\}\longleftarrow \boldsymbol{A}^{12,i}$ component of $\mathcal{S}_2;$
$\boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma}^{\mathrm{L1}})}\longleftarrow \boldsymbol{\Lambda}_{\mathrm{q}(\boldsymbol{\Sigma}^{\mathrm{L1}})}+\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)}\boldsymbol{\mu}^T_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)}+\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)}.$
For $j=1,\ldots,n_i$:

$\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L2}}_{ij})}\longleftarrow \boldsymbol{x}_{2,ij}$ component of $\mathcal{S}_2;\quad \boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L2}}_{ij})}\longleftarrow \boldsymbol{A}^{22,ij}$ component of $\mathcal{S}_2;$
$E_{\mathrm{q}}\{(\boldsymbol{\beta}-\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{\beta})})(\boldsymbol{u}^{\mathrm{L2}}_{ij}-\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L2}}_{ij})})^T\}\longleftarrow \boldsymbol{A}^{12,ij}$ component of $\mathcal{S}_2;$
$E_{\mathrm{q}}\{(\boldsymbol{u}^{\mathrm{L1}}_i-\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L1}}_i)})(\boldsymbol{u}^{\mathrm{L2}}_{ij}-\boldsymbol{\mu}_{\mathrm{q}(\boldsymbol{u}^{\mathrm{L2}}_{ij})})^T\}\longleftarrow \boldsymbol{A}^{12,i,j}$ component of $\mathcal{S}_2;$
continued on a subsequent page ...

Algorithm 3.3 continued. *This is a continuation of the description of this algorithm that commences on a preceding page.*

$$\lambda_{q(\sigma^2)} \longleftarrow \lambda_{q(\sigma^2)} + \left\|\boldsymbol{y}_{ij} - \boldsymbol{X}_{ij}\boldsymbol{\mu}_{q(\boldsymbol{\beta})} - \boldsymbol{Z}^{L1}_{ij}\boldsymbol{\mu}_{q(\boldsymbol{u}^{L1}_i)} - \boldsymbol{Z}^{L2}_{ij}\boldsymbol{\mu}_{q(\boldsymbol{u}^{L2}_{ij})}\right\|^2$$
$$+ \text{tr}(\boldsymbol{X}^T_{ij}\boldsymbol{X}_{ij}\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}) + \text{tr}((\boldsymbol{Z}^{L1}_{ij})^T\boldsymbol{Z}^{L1}_{ij}\boldsymbol{\Sigma}_{q(\boldsymbol{u}^{L1}_i)}) + \text{tr}((\boldsymbol{Z}^{L2}_{ij})^T\boldsymbol{Z}^{L2}_{ij}\boldsymbol{\Sigma}_{q(\boldsymbol{u}^{L2}_{ij})})$$
$$+ 2\text{tr}[(\boldsymbol{Z}^{L1}_{ij})^T\boldsymbol{X}_{ij}E_q\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\boldsymbol{\beta})})(\boldsymbol{u}^{L1}_i - \boldsymbol{\mu}_{q(\boldsymbol{u}^{L1}_i)})^T\}]$$
$$+ 2\text{tr}[(\boldsymbol{Z}^{L2}_{ij})^T\boldsymbol{X}_{ij}E_q\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\boldsymbol{\beta})})(\boldsymbol{u}^{L2}_{ij} - \boldsymbol{\mu}_{q(\boldsymbol{u}^{L2}_{ij})})^T\}]$$
$$+ 2\text{tr}[(\boldsymbol{Z}^{L1}_{ij})^T\boldsymbol{Z}^{L2}_{ij}E_q\{(\boldsymbol{u}^{L1}_i - \boldsymbol{\mu}_{q(\boldsymbol{u}^{L1}_i)})(\boldsymbol{u}^{L2}_{ij} - \boldsymbol{\mu}_{q(\boldsymbol{u}^{L2}_{ij})})^T\}];$$

$$\boldsymbol{\Lambda}_{q(\boldsymbol{\Sigma}^{L2})} \longleftarrow \boldsymbol{\Lambda}_{q(\boldsymbol{\Sigma}^{L2})} + \boldsymbol{\mu}_{q(\boldsymbol{u}^{L2}_{ij})}\boldsymbol{\mu}^T_{q(\boldsymbol{u}^{L2}_{ij})} + \boldsymbol{\Sigma}_{q(\boldsymbol{u}^{L2}_{ij})}.$$

$$\mu_{q(1/\sigma^2)} \longleftarrow \xi_{q(\sigma^2)}/\lambda_{q(\sigma^2)};$$

$$\boldsymbol{M}_{q((\boldsymbol{\Sigma}^{L1})^{-1})} \longleftarrow (\xi_{q(\boldsymbol{\Sigma}^{L1})} - q_1 + 1)\boldsymbol{\Lambda}^{-1}_{q(\boldsymbol{\Sigma}^{L1})}; \quad \boldsymbol{M}_{q((\boldsymbol{\Sigma}^{L2})^{-1})} \longleftarrow (\xi_{q(\boldsymbol{\Sigma}^{L2})} - q_2 + 1)\boldsymbol{\Lambda}^{-1}_{q(\boldsymbol{\Sigma}^{L2})};$$

$$\lambda_{q(a_{\sigma^2})} \longleftarrow \mu_{q(1/\sigma^2)} + 1/(\nu_{\sigma^2}s^2_{\sigma^2}); \quad \mu_{q(1/a_{\sigma^2})} \longleftarrow \xi_{q(a_{\sigma^2})}/\lambda_{q(a_{\sigma^2})};$$

$$\lambda_{q(\tau^2)} \longleftarrow \mu_{q(1/a_{\tau^2})} + \boldsymbol{\mu}^T_{q(\zeta)}\boldsymbol{\mu}_{q((\boldsymbol{\beta}^S)^2)}; \quad \mu_{q(1/\tau^2)} \longleftarrow \xi_{q(\tau^2)}/\lambda_{q(\tau^2)};$$

$$\lambda_{q(a_{\tau^2})} \longleftarrow \mu_{q(1/\tau^2)} + 1/s^2_{\tau^2}; \quad \mu_{q(1/a_{\tau^2})} \longleftarrow \xi_{q(a_{\tau^2})}/\lambda_{q(a_{\tau^2})};$$

$$\boldsymbol{g} \longleftarrow \tfrac{1}{2}\mu_{q(1/\tau^2)}\boldsymbol{\mu}_{q((\boldsymbol{\beta}^S)^2)};$$

If Laplace: $\boldsymbol{\mu}_{q(\zeta)} \longleftarrow \sqrt{\mathbf{1}/(2\boldsymbol{g})}$;

If Horseshoe: $\boldsymbol{\lambda}_{q(\zeta)} \longleftarrow \boldsymbol{\mu}_{q(a_\zeta)} + \boldsymbol{g}; \quad \boldsymbol{\mu}_{q(\zeta)} \longleftarrow \mathbf{1}/\boldsymbol{\lambda}_{q(\zeta)};$
$\boldsymbol{\lambda}_{q(a_\zeta)} \longleftarrow \boldsymbol{\mu}_{q(\zeta)} + \mathbf{1}; \quad \boldsymbol{\mu}_{q(a_\zeta)} \longleftarrow \mathbf{1}/\boldsymbol{\lambda}_{q(a_\zeta)};$

If NEG: $\boldsymbol{\lambda}_{q(\zeta)} \longleftarrow 2\boldsymbol{\mu}_{q(a_\zeta)}; \quad \boldsymbol{\mu}_{q(\zeta)} \longleftarrow \sqrt{\boldsymbol{\lambda}_{q(\zeta)}/(2\boldsymbol{g})}; \quad \boldsymbol{\mu}_{q(\mathbf{1}/\zeta)} \longleftarrow \mathbf{1}/\boldsymbol{\mu}_{q(\zeta)} + \mathbf{1}/(2\boldsymbol{\mu}_{q(a_\zeta)});$
$\boldsymbol{\lambda}_{q(a_\zeta)} \longleftarrow \boldsymbol{\mu}_{q(\mathbf{1}/\zeta)} + \mathbf{1}; \quad \boldsymbol{\mu}_{q(a_\zeta)} \longleftarrow (\lambda + 1)(\mathbf{1}/\boldsymbol{\lambda}_{q(a_\zeta)});$

$$\boldsymbol{\Lambda}_{q(A_{\boldsymbol{\Sigma}^{L1}})} \longleftarrow \text{diag}\{\text{diagonal}(\boldsymbol{M}_{q((\boldsymbol{\Sigma}^{L1})^{-1})})\} + \boldsymbol{\Lambda}_{A_{\boldsymbol{\Sigma}^{L1}}}; \quad \boldsymbol{M}_{q((A_{\boldsymbol{\Sigma}^{L1}})^{-1})} \longleftarrow \xi_{q(A_{\boldsymbol{\Sigma}^{L1}})}\boldsymbol{\Lambda}^{-1}_{q(A_{\boldsymbol{\Sigma}^{L1}})}.$$

$$\boldsymbol{\Lambda}_{q(A_{\boldsymbol{\Sigma}^{L2}})} \longleftarrow \text{diag}\{\text{diagonal}(\boldsymbol{M}_{q((\boldsymbol{\Sigma}^{L2})^{-1})})\} + \boldsymbol{\Lambda}_{A_{\boldsymbol{\Sigma}^{L2}}}; \quad \boldsymbol{M}_{q((A_{\boldsymbol{\Sigma}^{L2}})^{-1})} \longleftarrow \xi_{q(A_{\boldsymbol{\Sigma}^{L2}})}\boldsymbol{\Lambda}^{-1}_{q(A_{\boldsymbol{\Sigma}^{L2}})}.$$

Outputs: $\boldsymbol{\mu}_{q(\boldsymbol{\beta})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}, \left\{\boldsymbol{\mu}_{q(\boldsymbol{u}^{L1}_i)}, \boldsymbol{\Sigma}_{q(\boldsymbol{u}^{L1}_i)}, E_q\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\boldsymbol{\beta})})(\boldsymbol{u}^{L1}_i - \boldsymbol{\mu}_{q(\boldsymbol{u}^{L1}_i)})^T\} : 1 \le i \le m\right\},$

$\left\{\boldsymbol{\mu}_{q(\boldsymbol{u}^{L2}_{ij})}, \boldsymbol{\Sigma}_{q(\boldsymbol{u}^{L2}_{ij})}, E_q\{(\boldsymbol{\beta} - \boldsymbol{\mu}_{q(\boldsymbol{\beta})})(\boldsymbol{u}^{L2}_{ij} - \boldsymbol{\mu}_{q(\boldsymbol{u}^{L2}_{ij})})^T\} : 1 \le i \le m, 1 \le j \le n_i\right\},$

$\xi_{q(\sigma^2)}, \lambda_{q(\sigma^2)}, \xi_{q(a_{\sigma^2})}, \lambda_{q(a_{\sigma^2})}, \xi_{q(\tau^2)}, \lambda_{q(\tau^2)}, \xi_{q(a_{\tau^2})}, \lambda_{q(a_{\tau^2})},$

if Laplace: $\boldsymbol{\mu}_{q(\zeta)}$, if Horseshoe: $\left\{\boldsymbol{\lambda}_{q(\zeta)}, \boldsymbol{\lambda}_{q(a_\zeta)}\right\}$, if NEG: $\left\{\boldsymbol{\mu}_{q(\zeta)}, \boldsymbol{\lambda}_{q(\zeta)}, \boldsymbol{\lambda}_{q(a_\zeta)}\right\}$,

$\xi_{q(\boldsymbol{\Sigma}^{L1})}, \boldsymbol{\Lambda}_{q(\boldsymbol{\Sigma}^{L1})}, \xi_{q(\boldsymbol{\Sigma}^{L2})}, \boldsymbol{\Lambda}_{q(\boldsymbol{\Sigma}^{L2})}, \xi_{q(A_{\boldsymbol{\Sigma}^{L1}})}, \boldsymbol{\Lambda}_{q(A_{\boldsymbol{\Sigma}^{L1}})}, \xi_{q(A_{\boldsymbol{\Sigma}^{L2}})}, \boldsymbol{\Lambda}_{q(A_{\boldsymbol{\Sigma}^{L2}})}.$

3.6 Numerical Investigations on Simulated Data

We now discuss the results of a simulation study conducted to assess:

1. the accuracy of the optimal q-density approximations compared to the marginal posterior density functions obtained via MCMC when global-local priors are specified;

2. fixed effects selection performances via the SAVS procedure for effectively discriminating the relevant fixed effects from those being irrelevant;

3. computational timings and memory storage requirements for both naïve and streamlined algorithm implementations, especially when the number of model parameters increases.

Notice that the accuracy and variable selection performances are not affected by the use of streamlined updates in place of naïve counterparts, given that both the implementations converge to the same solution.

To allow for maximal speed, both the MFVB algorithms and corresponding MCMC sampling schemes were implemented in C++ employing the Armadillo library and executed in R using the RcppArmadillo package (Eddelbuettel and Sanderson, 2014). See Appendix A.3 for further details.

The simulation study focused on three-level random effect models, which give rise to more complex three-level sparse matrix structures. We simulated 50 datasets according to model specification (3.14) with $m = 100$ groups, each with $n_i = 15$ sub-groups, each having $o_{ij} = 20$ units, for $1 \leq i \leq 100, 1 \leq j \leq 15$. We included a random intercept and a random slope for both the group and sub-group levels ($q_1 = q_2 = p_{\text{R}} = 2$) with true parameter values being $\boldsymbol{\beta}^{\text{R}} = (0.58, 1.98)^T$, and $p_{\text{A}} = 3$ additional fixed effects having true parameter values $\boldsymbol{\beta}^{\text{A}} = (0.7, -0.9, 1.8)^T$. We also considered a sparse design setting with $p_{\text{S}} = 50$ fixed effects such that 40 of them were assumed to be irrelevant and hence the corresponding parameters had true value equal to zero, while the remaining 10 parameters were $\beta_1^{\text{S}} = 1.91, \beta_7^{\text{S}} = 1.96, \beta_{10}^{\text{S}} = -0.10, \beta_{18}^{\text{S}} = 1.62, \beta_{24}^{\text{S}} = -1.45, \beta_{25}^{\text{S}} = -1.53, \beta_{36}^{\text{S}} = 0.24, \beta_{37}^{\text{S}} = 1.76, \beta_{45}^{\text{S}} = 1.79$ and $\beta_{49}^{\text{S}} = -0.15$. The true variance parameter σ^2 was fixed to 0.7. The *true* random effects vectors $\boldsymbol{u}_i^{\text{L1}}$ and $\boldsymbol{u}_{ij}^{\text{L2}}$ were generated independently from $\text{N}(\mathbf{0}, \boldsymbol{\Sigma}^{\text{L1}})$ and $\text{N}(\mathbf{0}, \boldsymbol{\Sigma}^{\text{L2}})$ distributions, having

$$\boldsymbol{\Sigma}^{\text{L1}} = \begin{bmatrix} 0.42 & -0.09 \\ -0.09 & 0.52 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}^{\text{L2}} = \begin{bmatrix} 0.80 & -0.24 \\ -0.24 & 0.75 \end{bmatrix}.$$

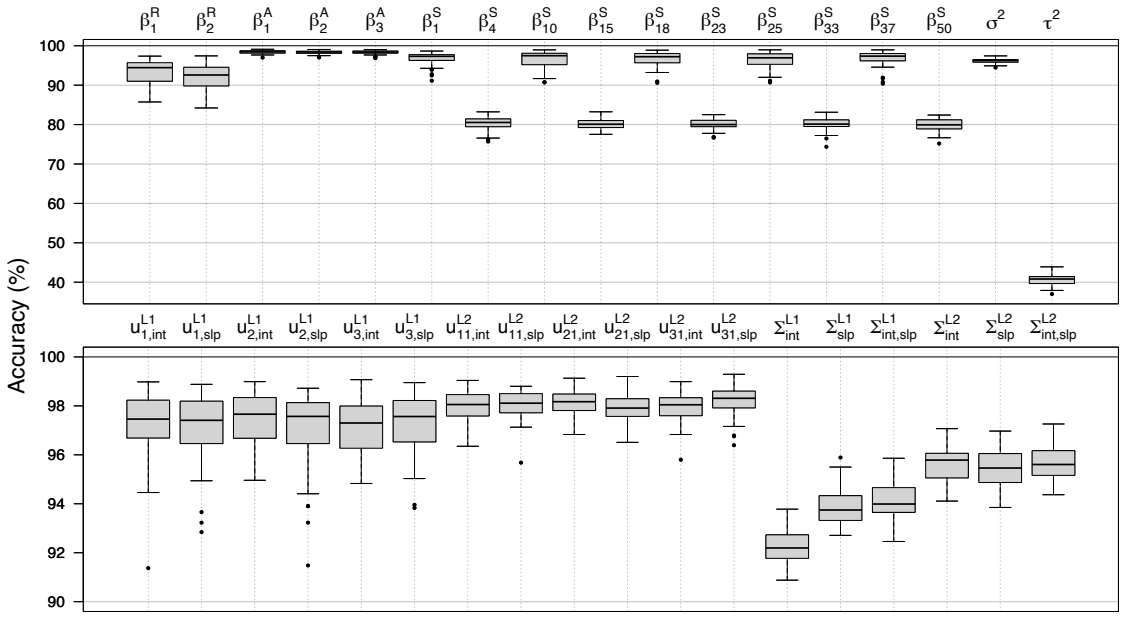

FIGURE 3.2: *Side-by-side boxplots of the accuracy scores from the simulation study for a selection of model parameters and random effects. Outliers are displayed as solid points.*

For each data replication, the slope-associated column of $\boldsymbol{X}^{\mathrm{R}}$ was generated from a standard Gaussian distribution, while the rows of $\boldsymbol{X}^{\mathrm{A}}$ and $\boldsymbol{X}^{\mathrm{S}}$ were generated from two multivariate Gaussian distributions having zero mean vector and covariance matrices generated from Wishart$(p_{\mathrm{A}}, \boldsymbol{I})$ and Wishart$(p_{\mathrm{S}}, \boldsymbol{I})$ distributions, respectively. This strategy produces covariates with different variability and non-zero correlations, better mimicking a real-data scenario. The setting embeds a fixed effects design matrix $\boldsymbol{X}$ of size 30,000 $\times$ 55, and a sparse random effects design matrix $\boldsymbol{Z}$ of size 30,000 $\times$ 3,200 with 96 millions cells, of which 99.875% are zeros.

We proceed fitting model (3.14) for each data replication using diffuse priors with hyperparameters $\boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{R}}} = \boldsymbol{\mu}_{\boldsymbol{\beta}^{\mathrm{A}}} = \boldsymbol{0}$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{R}}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{\mathrm{A}}} = 10^{10}\boldsymbol{I}$, $\nu_{\sigma^2} = 1$, $\nu_{\boldsymbol{\Sigma}^{\mathrm{L1}}} = \nu_{\boldsymbol{\Sigma}^{\mathrm{L2}}} = 2$, $s_{\sigma^2} = s_{\boldsymbol{\Sigma}^{\mathrm{L1}},1} = s_{\boldsymbol{\Sigma}^{\mathrm{L1}},2} = s_{\boldsymbol{\Sigma}^{\mathrm{L2}},1} = s_{\boldsymbol{\Sigma}^{\mathrm{L2}},2} = 10^5$. Without loss of generality, an uninformative Horseshoe prior for all the elements of $\boldsymbol{\beta}^{\mathrm{S}}$ have been specified, with $s_{\tau^2} = 10^5$ to limit prior information about the global degree of sparsity. Along with the simulation study, the MFVB approximations were obtained by running 200 iterations of the streamlined or naïve algorithms. In comparison, the MCMC samplings were performed using a warmup of length 5,000 followed by 25,000 iterations, to which we applied a thinning procedure of size 5 for preventing chain autocorrelation.

3.6.1 Accuracy Assessment

We measured the quality of the variational approximations using the accuracy index (1.25). Figure 3.2 shows the accuracy results for the 50 data replicates. The boxplots refer to all the entries of $\boldsymbol{\beta}^{\mathrm{R}}$ and $\boldsymbol{\beta}^{\mathrm{A}}$, to 10 elements of $\boldsymbol{\beta}^{\mathrm{S}}$ chosen such that 5 of them have non-zero values (β^{S}_1, β^{S}_{10}, β^{S}_{18}, β^{S}_{25} and β^{S}_{37}) and 5 are null (β^{S}_4, β^{S}_{15}, β^{S}_{23}, β^{S}_{33} and β^{S}_{50}), the intercept- and slope-associated elements of $\boldsymbol{u}^{\mathrm{L1}}_i$ and $\boldsymbol{u}^{\mathrm{L2}}_{ij}$, for $1 \leq i \leq 3$ and $j = 1$, the entries of $\boldsymbol{\Sigma}^{\mathrm{L1}}$ and $\boldsymbol{\Sigma}^{\mathrm{L2}}$, σ^2, and τ^2. The intercept- and slope-associated parameters are identified by the subscripts *int* and *slp*, respectively.

Variational approximations showed high accuracy scores for all the model parameters considered and across the different data replications. The accuracies for β^{R}_1 and β^{R}_2 were slightly affected by inappropriate convergence of their respective MCMC chains, which may be resolved employing hierarchical centering methods (see Gelfand *et al.*, 1995 and Section 2.3 of Zhao *et al.*, 2006). All the fixed effects subject to selection and having non-zero true values exhibited accuracy scores greater than 90%. In contrast, those having true values equal to zero had lower accuracy scores between 75% and 85% due to the spiky marginal posterior densities being approximated by Gaussian variational densities. All the other fixed effects parameters, random effects and variance parameters showed accuracy scores greater than 90%. The accuracy scores of the global variance parameter τ^2 were under 50%. Despite not being directly shown, all the $\mathfrak{q}^*(\zeta_h)$'s had accuracy scores between 75% and 80%. Similar results were obtained for the alternative global-local prior specifications.

Figure 3.3 displays the MFVB and MCMC approximate posterior densities, together with the associated accuracy scores, obtained for the first data replication of the simulation study. These plots visually assess the quality of the approximation and show that the approximate posterior density functions are generally concentrated around the true parameter values. Notice that the global-local prior shrinks the MCMC marginal posterior densities of the β^{S}-parameters having true null value towards zero. The corresponding optimal $\mathfrak{q}$-densities are Gaussian and, although they cannot capture peak and tail behaviors, provide satisfactory approximations. Notice also that the τ^2 approximate posterior density is concentrated around a very small value, as expected for the sparse covariate setting we considered in the simulation study.

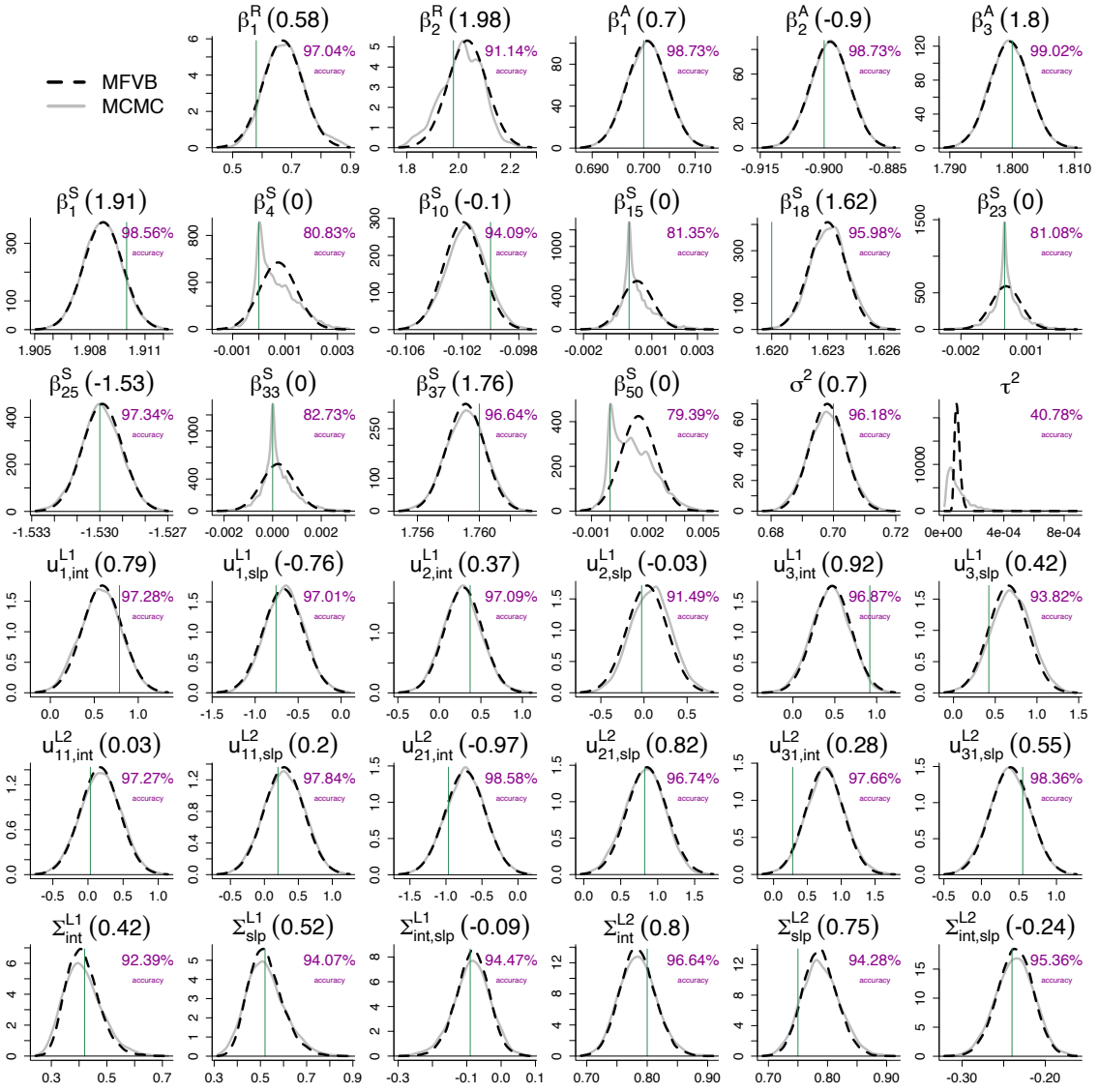

FIGURE 3.3: *Approximate posterior density functions of some of the three-lever random effects model parameters obtained from the first replication of the simulation study. Each plot shows the optimal approximate posterior density function* $\mathfrak{q}^*(\theta)$ *obtained via MFVB (black dashed curves) and the MCMC-based* $\mathfrak{p}(\theta|\boldsymbol{y})$ *densities (grey curves). A vertical line indicates the parameter true value. The percentages of accuracy are also provided.*

3.6.2 Fixed Effects Selection Assessment

We assessed fixed effects selection performances by running Algorithm 3.3 on the same 50 data replications, also experimenting the other two global-local priors considered in this work: the Laplace and Normal-Exponential-Gamma with $\lambda = 0.25$. We also considered the Gaussian prior specification described in Nolan *et al.* (2020) with hyperparameters $\boldsymbol{\mu}_{\boldsymbol{\beta}^S} = \mathbf{0}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\beta}^S} = 10^{10}\boldsymbol{I}$. Then, for each data replication and each of the four different priors considered, we used the approximating densities corresponding to fixed effects subject to selection to perform the SAVS procedure of

Algorithm 3.1. The SAVS procedure have also been employed for the approximate posteriors obtained from the Gaussian prior in order to compare with a prior that does not belong to the global-local family.

Let TP (true positives) denote the number of selected fixed effects with true value different from zero, and TN (true negatives) denote the number of unselected irrelevant fixed effects having true value equal to zero. We measured the fixed effects selection performance for each prior considered using the F_1-score (van Rijsbergen, 2004):

$$F_1 \equiv \left(\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \right) \times 100\%, \tag{3.19}$$

with precision $\equiv$ TP$/$(TP $+$ FP) and recall $\equiv$ TP$/$(TP $+$ FN), where FP and FN denote the number of false positives (incorrectly selected fixed effects) and false negatives (relevant fixed effects that have not been selected), respectively. This index takes values between 0% and 100%, with higher values to be preferred.

The median F_1-score over the 50 data replicate was 63.5% (1st quartile: 43.5%; 3rd quartile: 94.2%) for the Gaussian prior and 95.24% (1st quartile: 80%; 3rd quartile: 100%) for the Laplace prior. Both the Horseshoe and Negative-Exponential-Gamma priors exhibited a F_1-score equal to 100% for all the 50 data replications of the simulation study, meaning that perfect selection (TP $=$ 10 and TN $=$ 40) was always achieved. From this simplified simulated sparse data scenario, it is apparent that global-local priors effectively provide better variable selection performances than the Gaussian prior. The lower performances of the Gaussian and Laplace priors are due to the SAVS procedure being applied to optimal approximate q-densities that have not been properly shrunk towards zero, and so irrelevant fixed effects tend to be selected. Nonetheless, all the four different priors gave FP $=$ 0, meaning that they did not incorrectly select irrelevant fixed effects.

3.6.3 Speed and Memory Saving Assessment

The streamlined variational inference has been conceived to obtain efficient implementations of variational algorithms. Hence, the assessment of speed and memory savings is another important aspect. We assessed speed and memory performances of the streamlined variational Algorithm 3.3 and its naïve counterpart, whose updates are described in Section 3.5.1. Four different group numbers and three different lengths for the vector of fixed effects to be selected were considered, namely $m \in \{10, 50, 100, 200\}$ and $p_S \in \{25, 100, 200\}$, aiming to explore the scalability of the streamlined methodology to high dimensions. For each combination

		Total runtime of the algorithm (seconds)				Total size of the required data inputs (megabytes)		
m	p_S	*Streamlined MFVB*	*Naïve MFVB*	$\frac{\text{Naïve MFVB}}{\text{Streamlined MFVB}}$	*MCMC*	*Streamlined MFVB*	*Naïve MFVB*	$\frac{\text{Naïve MFVB}}{\text{Streamlined MFVB}}$
10	25	$0.70_{(0.06)}$	$3.86_{(0.50)}$	5.54	$13.03_{(0.96)}$	$1.39_{(0.07)}$	$11.06_{(1.00)}$	7.97
	100	$3.90_{(0.23)}$	$6.51_{(0.82)}$	1.67	$46.99_{(1.47)}$	$3.68_{(0.24)}$	$12.73_{(1.34)}$	3.46
	200	$9.71_{(1.22)}$	$12.09_{(2.04)}$	1.24	$176.22_{(9.37)}$	$6.60_{(0.52)}$	$15.09_{(1.89)}$	2.29
50	25	$3.29_{(0.11)}$	$365.07_{(37.10)}$	111.01	$63.28_{(3.76)}$	$6.69_{(0.20)}$	$240.40_{(13.92)}$	35.91
	100	$19.66_{(0.87)}$	$415.29_{(51.66)}$	21.12	$138.42_{(5.72)}$	$18.37_{(0.81)}$	$254.04_{(20.84)}$	13.83
	200	$49.25_{(1.47)}$	$501.33_{(39.76)}$	10.18	$305.25_{(6.51)}$	$34.00_{(1.04)}$	$269.34_{(14.90)}$	7.92
100	25	$6.77_{(0.32)}$	$2877.33_{(177.77)}$	424.72	$151.05_{(10.48)}$	$13.56_{(0.34)}$	$967.34_{(42.15)}$	71.33
	100	$40.42_{(1.66)}$	$3172.66_{(116.83)}$	78.50	$266.71_{(10.06)}$	$36.77_{(1.29)}$	$1010.25_{(26.94)}$	27.47
	200	$97.81_{(2.07)}$	$3403.55_{(204.38)}$	34.80	$494.15_{(9.34)}$	$67.53_{(1.31)}$	$1013.90_{(48.67)}$	15.01
200	25	$13.30_{(0.30)}$	> 5 hours	> 1355	$328.87_{(20.04)}$	$26.90_{(0.56)}$	$3817.06_{(113.85)}$	141.88
	100	$79.97_{(1.18)}$	> 5 hours	> 225	$578.88_{(13.45)}$	$73.74_{(1.01)}$	$3941.15_{(97.90)}$	53.45
	200	$197.20_{(4.13)}$	> 5 hours	> 95	$904.64_{(15.77)}$	$135.84_{(2.73)}$	$3991.66_{(102.48)}$	29.38

TABLE 3.2: *Average (standard deviation of) elapsed computing times in seconds and average (standard deviation of) total size of required data inputs in megabytes for fitting model* (3.14) *with three-level random effects specification and p_S fixed effects having Horseshoe prior. Results are shown for different group sizes m and different values for p_S.*

of m and p_S, we simulated 10 data replications from model (3.14) with a random intercept and one slope for a single continuous predictor, choosing the sub-group dimensions n_i uniformly on the discrete set $\{10, \ldots, 20\}$ and the sub-group specific unit dimensions o_{ij} uniformly on the discrete set $\{20, \ldots, 30\}$. This setting allows testing models with heterogeneous dimensions, although sharing the same number of groups m. We considered again a Horseshoe prior for the fixed effects subject to selection, and all the other model dimensions and hyperparameters were the same as those described before.

For each simulated dataset, we collected the computational timings and the total size of the inputs data required for performing both the streamlined and naïve algorithms. We also ran MCMC and recorded its computational timings, although MFVB and MCMC do not admit a genuine comparison as already discussed in Section 1.5. Moreover, our efficient implementation of MCMC is based on independent resampling from the full conditional densities of $\boldsymbol{\beta}$, $\boldsymbol{u}_i^{\text{L1}}$ and $\boldsymbol{u}_{ij}^{\text{L2}}$, whereas MFVB uses a joint approximation $\mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u})$ for all these vectors. Nonetheless, we also report MCMC timings to provide intuition on the computational effort required for sampling from the posterior distribution when m and p_S increase.

The tabulated results are shown in Table 3.2. The *ratio* columns help to understand the gain obtained employing the streamlined MFVB methodology over its naïve counterpart. For increasing m, streamlined MFVB reached convergence faster than naïve MFVB. Notice that in the biggest scenario under examination (last row of the table), the streamlined MFVB algorithm ran in less than 4 minutes on average,

while naïve MFVB required more than 5 hours. Bigger scenarios are computationally demanding for the naïve implementation and may fail to run due to excessive storage demand, whilst we did not experience these issues with Algorithm 3.3. Similar comments apply to the huge saving of memory allocation for the required input data provided by the streamlined MFVB implementation.

Conditionally on the same m scenario, the speed gain and memory saving obtained by streamlining the MFVB updates decrease as p_S increases. This comes with no surprise as the work of Nolan and Wand (2020) accounts for sparse structures of $\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})}$ when m becomes larger and larger and p is assumed to be relatively very small. Having a closer look at its naïve update expression (3.18), we can conceptually subdivide it into

$$\boldsymbol{\Sigma}_{q(\boldsymbol{\beta},\boldsymbol{u})} \longleftarrow \left(\begin{bmatrix} \boldsymbol{A}_{\boldsymbol{\beta}} & \boldsymbol{A}_{\boldsymbol{\beta},\boldsymbol{u}} \\ \boldsymbol{A}_{\boldsymbol{u},\boldsymbol{\beta}} & \boldsymbol{A}_{\boldsymbol{u}} \end{bmatrix} \right)^{-1},$$

where $\boldsymbol{A}_{\boldsymbol{\beta}}$ is a full squared matrix of dimension p referring to the fixed effects component, $\boldsymbol{A}_{\boldsymbol{\beta},\boldsymbol{u}}$ is a full matrix of dimension $p \times (q_1 m + q_2 \sum_{i=1}^{m} n_i)$ and $\boldsymbol{A}_{\boldsymbol{u}}$ is a sparse squared matrix of dimension $q_1 m + q_2 \sum_{i=1}^{m} n_i$ referring to the random effects component. The matrix $\boldsymbol{A}_{\boldsymbol{\beta}}$ exactly corresponds to the $\boldsymbol{A}_{11}$ matrix of dimension $p \times p$ defined in Result 3.2 and refers to the sub-block which does not provide any sparse structure induced by the three-level random effects specification. The SolveThreeLevelSparseMatrix routine makes use of this matrix in many multiplications with the sub-blocks of $\boldsymbol{A}_{\boldsymbol{\beta},\boldsymbol{u}}$ and $\boldsymbol{A}_{\boldsymbol{u}}$, increasing the overall runtime as p gets bigger and bigger. Keeping m fixed and increasing the dimension of p_S (and p by consequence) makes the sub-block $\boldsymbol{A}_{\boldsymbol{\beta}}$ larger and larger with respect to $\boldsymbol{A}_{\boldsymbol{u}}$, and overwhelms the computational gain obtained by efficiently accounting for its sparse structure. This is apparent by the $(m = 10, p_S = 200)$ scenario, for which streamlined MFVB provides no significative advantages over the naïve implementation in terms of computational runtime. In general, streamlined MFVB computations and associated algorithms proposed in this chapter enhance their computational benefits when p is limited in comparison with m. Similar comments apply also to the two-level random effects specification.

We conclude the discussion by noticing that our methodology takes the approximate joint posterior dependence between the fixed effects and random effects parameters into account through the multivariate Gaussian approximating density

$\mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u})$. This choice allows to better capture the posterior covariance structure between $\boldsymbol{\beta}$ and $\boldsymbol{u}$ and ensures better results in terms of approximation accuracy. Nevertheless, alternative and less restrictive factorizations can be considered, especially if the quality of the approximation can be sacrificed and faster algorithms are desired. For instance, the additional factorization $\mathfrak{q}(\boldsymbol{\beta}, \boldsymbol{u}) = \mathfrak{q}(\boldsymbol{\beta}^{\mathrm{R}}, \boldsymbol{u})\mathfrak{q}(\boldsymbol{\beta}^{\mathrm{A}}, \boldsymbol{\beta}^{\mathrm{S}})$ could remarkably speed up the computations if p_{R} is small, regardless of the size of p_{A} and p_{S}. For implementing this and any other additional independence constraints, mean-field restrictions such as (3.16) and the algorithms proposed in this chapter need to be modified accordingly.

3.7 Application to Data from a Perinatal Study

We present an application of the methodology and algorithms proposed in Section 3.5 to the National Collaborative Perinatal Project data (Klebanoff, 2009), a multisite prospective cohort study that took place in the United States of America between 1959 and 1974. This study was designed to identify the effects of complications during pregnancy or the perinatal period on birth and child outcomes. The data are publicly available from the U.S. National Archives with identifier 606622. Many online resources have already employed this dataset or subset of it for many analyses. Over the years, it has become a high-quality reference for biomedical and behavioral research in many areas such as obstetrics, perinatology, pediatrics, and developmental psychology.

The same data were examined in Nolan *et al.* (2020) and we expressly account for the same model specification to experiment a suitable fixed effects selection for a moderately large set of regressors that have been excluded from their analysis and that may have a relevant impact onto explaining the response variable. A full-blown analysis goes beyond the scope of this paper, and we focused on predicting the *height-for-age z-score* for 37,257 infants followed longitudinally over their first year of life, following indications from Taylor (1980). The height-for-age z-score is a standardized measure of the World Health Organization for children's height after accounting for age. Notable discrepancies from this index standard reference values are considered an alarm signal for malnutrition symptoms.

We performed a Bayesian analysis that accounts for the heterogeneity of the evolution of such index across infants. Our analysis was performed through a two-level random effects linear model having a random intercept, and linear and quadratic slopes ($q = 3$) to account for the quadratic evolution of that score over time. All

the fixed effects regressors, excluding the intercept, age of the infant (in days since birth) and its square, were subject to selection; these included characteristics of the infant at birth (e.g. weight, length, head circumference, sex, and Apgar scores), and characteristics of the mother, father and family. We also accounted for possible interactions between some infant characteristics and sex, for a total of 38 candidate predictors subject to selection.

The model we fitted respects the general specification (3.14) and can be expressed for the generic ith infant ($1 \leq i \leq 37,257$) as follows:

$$
\begin{aligned}
&\boldsymbol{y}_i|\boldsymbol{\beta}, \boldsymbol{u}_i, \sigma^2 \overset{\text{ind}}{\sim} \mathrm{N}(\boldsymbol{X}_i^{\mathrm{R}}\boldsymbol{\beta}^{\mathrm{R}} + \boldsymbol{X}_i^{\mathrm{S}}\boldsymbol{\beta}^{\mathrm{S}} + \boldsymbol{Z}_i\boldsymbol{u}_i, \sigma^2\boldsymbol{I}), \quad \boldsymbol{u}_i|\Sigma \overset{\text{ind}}{\sim} \mathrm{N}_3(\boldsymbol{0}, \boldsymbol{\Sigma}),\\
&\left[\begin{array}{c} \boldsymbol{\beta}^{\mathrm{R}} \\ \boldsymbol{\beta}^{\mathrm{S}} \end{array}\right] \Bigg| \boldsymbol{\zeta}, \tau^2 \sim \mathrm{N}\left(\left[\begin{array}{c} \boldsymbol{0}_3 \\ \boldsymbol{0}_{38} \end{array}\right], \left[\begin{array}{cc} 10^{10}\boldsymbol{I}_3 & \boldsymbol{O} \\ \boldsymbol{O} & \tau^2\,\mathrm{diag}(\boldsymbol{\zeta})^{-1} \end{array}\right]\right),\\
&\sigma^2|a_{\sigma^2} \sim \text{Inverse-}\chi^2(1, 1/a_{\sigma^2}), \quad a_{\sigma^2} \sim \text{Inverse-}\chi^2(1, 10^{-10}),\\
&\boldsymbol{\Sigma}|\boldsymbol{A}_{\boldsymbol{\Sigma}} \sim \text{Inverse-G-Wishart}(G_{\text{full}}, 6, \boldsymbol{A}_{\boldsymbol{\Sigma}}^{-1}),\\
&\boldsymbol{A}_{\boldsymbol{\Sigma}} \sim \text{Inverse-G-Wishart}(G_{\text{diag}}, 1, 2 \times 10^{-10}\boldsymbol{I}_3),\\
&\zeta_h|a_{\zeta_h} \overset{\text{ind}}{\sim} \begin{cases} \text{Inverse-}\chi^2(2,1) & \text{for a Laplace prior} \\ \text{Gamma}(1/2, a_{\zeta_h}) & \text{for a Horseshoe prior} \\ \text{Inverse-}\chi^2(2, 2a_{\zeta_h}) & \text{for a NEG prior,} \end{cases} \qquad (3.20)\\
&a_{\zeta_h} \overset{\text{ind}}{\sim} \begin{cases} - & \text{for a Laplace prior} \\ \text{Gamma}(1/2, 1) & \text{for a Horseshoe prior} \\ \text{Gamma}(\lambda, 1) & \text{for a NEG prior,} \end{cases}\\
&\qquad\qquad \text{for } 1 \leq h \leq 38,\\
&\tau^2|a_{\tau^2} \sim \text{Inverse-}\chi^2(1, 1/a_{\tau^2}), \quad a_{\tau^2} \sim \text{Inverse-}\chi^2(1, 10^{-10}).
\end{aligned}
$$

For the ith infant, o_i time-point measurements were recorded, ranging from one to four in number. The $\boldsymbol{X}_i^{\mathrm{R}}$ matrix has size $n_i \times 3$ with the first column being a vector of ones, the second one consisting of the time-point measurements for the ith infant and the third column containing the square of the elements of the second one. We set $\boldsymbol{X}_i^{\mathrm{A}} = \boldsymbol{O}$, while the $\boldsymbol{X}_i^{\mathrm{S}}$ matrix of size $n_i \times 38$ consisted of all the considered predictors subject to selection. Moreover, $\boldsymbol{Z}_i = \boldsymbol{X}_i^{\mathrm{R}}$ by definition. Uninformative priors were placed over all the model parameters. We fitted the model using the three different global-local priors treated in this work and the Gaussian prior for $\boldsymbol{\beta}^{\mathrm{S}}$ considered in Nolan *et al.* (2020).

Streamlined MFVB and MCMC were used for model fitting. The former was performed running Algorithm 3.2 and stopping it after 200 iterations. The latter was performed running 25,000 iterations, to which a thinning procedure of size 5 was applied after discarding 5,000 burnin iterations. The whole input data required approximately 200 megabytes of memory storage, while a naïve MFVB procedure would necessitate several gigabytes of memory to store the $\boldsymbol{Z}$ matrix, which is composed of 11,693,705,562 cells of which the 99.997% are zeros. All the covariates excluding the binary ones were standardized, and the estimates were rescaled to the original scale before presenting the results. The streamlined MFVB algorithms took 2 to 3 minutes to run for each different global-local prior specification, while the associated MCMC samplers always took more than 35 minutes.

We omit the presentation of model interpretation, goodness-of-fit analysis, accuracy of the approximations, convergence of the MCMC chains, and visualization of the fitted height-for-age z-score trajectories over time. Instead, in Figure 3.4 we present 90% high posterior density credible intervals for all the fixed effects subject to selection. The thicker lines represent the intervals obtained from the MCMC posterior samples through the `emp.hpd` function of the `TeachingDemos` package (Snow, 2020), while the thinner lines represent those calculated from the determined optimal MFVB approximate posteriors. The two lines are superimposed to facilitate immediate comparison for each different global-local prior specification.

Overall, MFVB provided very accurate high posterior density credible intervals when compared to MCMC. The MCMC chain associated with the `birth length` fixed effect showed some convergence problems, probably due to its moderate correlation with the response variable, and this reduced the overlap with MFVB. Notably, the Laplace prior produced results very similar to those provided by the Gaussian prior specification. On the other hand, both the Horseshoe and Normal Exponential Gamma priors seem to have effectively shrunk most of the fixed effects towards zero, especially those associated with dummy variables.

The SAVS procedure applied to the approximate posterior densities classifies the following fixed effects as relevant: the indicator that the infant is male, infant's year of birth, birth length, birth head circumference, the Apgar score assessed 5 minutes after the infant's birth, and the mother's height. We have little knowledge of the considered topic to confirm whether these results align with literature references. However, we are confident that sex, length of both the infant and the mother and some additional infant-related covariates are effectively relevant for predicting the height-for-age z-score.

FIGURE 3.4: *The 90% high posterior density credible intervals for the fixed effects subject to selection in the real data application. The four different priors for* $\boldsymbol{\beta}^S$ *are represented with different colors. For each fixed effect, the thicker lines correspond to the intervals obtained from the MCMC approximate marginal posterior densities, while the thinner lines represent those obtained from the streamlined MFVB approximating densities.*

Some fixed effects such as the ethnicity of the mother, the place where she conceived, her marital status and information on previous pregnancies were not indicated as relevant by the SAVS procedure, albeit some of their associated high posterior density credible intervals are far from zero. Notice also that for some fixed effects, global-local priors drag the intervals towards the origin, with different intensities probably depending on the variability and correlation of the covariates. More ad-hoc analyses are firmly suggested, although these go beyond the scope of the current work.

3.8 Concluding Remarks

In this chapter, we developed streamlined mean-field variational Bayes procedures for Gaussian response linear mixed models having nested random effects structures and admitting fixed effects prior specifications alternative to the classical Gaussian one. The priors we considered are amenable to automated and hyperparameter-free fixed effects selection procedures. Simulated and real data examples showed how streamlined variational inference could provide impressive benefits in terms of computational time and memory saving compared to inefficient implementations of variational approximations. Albeit the marginal posterior densities of fixed effects subject to selection are approximated by bell-shaped curves, our studies showed high performances of the automated selection procedure. It is also worth mentioning that the more restrictive mean-field approximations discussed at the end of Section 3.6 can sensibly improve the benefit of streamlined variational inference for large p scenarios and, in general, when speed is more important than approximation accuracy.

Numerous ramifications of the proposed methodology can be envisaged. These include the extension to generalized linear mixed models (e.g., for binary or Poisson response variables) or the treatment of models with unit-specific errors, heteroskedastic covariance structures for groups and sub-groups, higher levels of nesting, or crossed random effects.

Regarding the selection of fixed effects, alternative shrinkage priors belonging to the global-local family can be accounted for with minor modifications to the proposed algorithms. Following the prescriptions of Neville *et al.* (2014), it is possible to study global-local prior specifications that are not based on auxiliary variables and investigate whether this could lead to better approximation accuracies, although at the cost of more computationally intensive variational updates. We also mention the possibility of admitting spike-and-slab prior formulations following, for example, a variational inference approach similar to that of Carbonetto and Stephens (2012). The main drawback with spike-and-slab priors is that more involved algebra is required for implementing streamlined variational inference procedures.

Chapter 4

GVA for Nonstationary Gaussian Process Regression

4.1 Introduction

Gaussian processes (Stein, 1999; Rasmussen and Williams, 2006) are a powerful yet practical nonparametric Bayesian method for solving supervised learning problems such as nonlinear regression or classification. Their extensive use is due to their simplicity, flexibility and substantial theoretical support (e.g. Choi and Schervish, 2007; van der Vaart and van Zanten, 2008; Castillo, 2008; Koepernik and Pfaf, 2021) and successful applications include time-series modeling (e.g. Roberts *et al.*, 2013), spatial statistics (e.g. Cressie, 1993), geostatistics (e.g. Journel and Huijbregts, 1978), metereology (e.g. Thompson, 1956), health monitoring (e.g. Stegle *et al.*, 2008) and reinforcement learning (e.g. Kuss and Rasmussen, 2004) among others.

In this chapter, we take a *machine-learning perspective* and focus on Gaussian process regression models (Rasmussen and Williams, 2006), namely regression models in which Gaussian processes are employed as prior distributions over the unknown latent regression function. Standard machine learning literature correctly treats them as *nonparametric* Bayesian probabilistic regression models, because they embed a nonparametric prior distribution over the set of latent regression functions and Bayes theorem updates it with observed data into the posterior predictive distribution (O'Hagan, 1978). Nonetheless, this usually generates some confusion since their specification depends upon hyperparameters characterizing their mean and covariance functions, which needs to be estimated appropriately with either a *maximum-likelihood* or a Bayesian approach. Consistently with the content of this PhD thesis, we adopt a *fully-Bayesian* inferential formulation, the name underlining

that we perform Bayesian inference over a nonparametric Bayesian model and, as such, treat probabilistically model (hyper)parameters assigning proper prior distributions.

Regardless of the inferential strategy adopted, a typical approach to Gaussian process regression modeling is accounting for stationary prior specifications, for which a wide variety of functional patterns are encompassed within covariance function specifications depending on few parameters. This makes the estimation procedure computationally feasible as long as the observed data reflect stationary behaviors. Nevertheless, real data are well known to often exhibit irregular or unstable patterns which may compromise inferential conclusions if incorrectly modeled assuming stationarity a-priori. Adopting nonstationary Gaussian process prior specifications is currently a prolific area of research and many contributions emerged over the years: in this chapter we focus on convolution methods (Higdon *et al.*, 1999) from which Paciorek (2003) and Paciorek and Schervish (2006) derived interesting nonstationary covariance functions. Their intuition extended independent work of Gibbs (1997) by introducing nonstationarity accounting for predictor-dependent length-scale parameters, and we model their unknown relationship with a stationary Gaussian process prior following the characterization of deep Gaussian processes proposed by Dunlop *et al.* (2018).

MCMC sampling procedures for determining the posterior distribution of Gaussian process regression models accounting for such nonstationary covariance functions are summarized in Paciorek (2003). However, they are well known to be computationally demanding and the produced chain often exhibit moderate autocorrelation. Instead, we investigate GVA approximations to the posterior distribution employing a parsimonious factor covariance structure described by Ong *et al.* (2018), which is particularly suitable when the model parameter dimension is high. A simulation study allows us to describe the broad flexibility of nonstationary data patterns that are accommodated within our model specification and how it affords better predictions compared to stationary Gaussian process regression modeling. Drawbacks and computational limits of our variational approximation approach are summarized at the end of the chapter, together with possible solutions currently under development and investigation.

4.2 Gaussian Process Regression in a Nutshell

Many formulations and definitions for Gaussian processes are present in literature, depending on the applicative context of interest, the type of statistical problem for which they are employed, and the required level of mathematical abstraction. In order to proceed with a clear and shared notational convention throughout this chapter, we start by recalling in this section the main elements of Gaussian process regression from a so-called *function-space view*, which constitutes the main source of interest for this chapter, and to describe how a fully-Bayesian inferential treatment is handled. For a more in-depth exposure, we refer to Rasmussen and Williams (2006).

4.2.1 Background

Let we assume n predictor-response pairs $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathbb{R}$ have been observed, $\mathcal{X} \subseteq \mathbb{R}^d$, and that it is of interest to investigate the unknown regression function $f : \mathcal{X} \longrightarrow \mathbb{R}$ that relates them. A *Gaussian process regression (GPR)* model assumes each response y is independently generated from:

$$y = f(\boldsymbol{x}) + \epsilon, \quad \epsilon \sim \mathrm{N}(0, \sigma^2),$$

where $\sigma^2 > 0$ and

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \quad \text{for any} \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}. \tag{4.1}$$

Expression (4.1) yields a *Gaussian process (GP)* prior over f having *mean function* $m : \mathcal{X} \longrightarrow \mathbb{R}$ and *covariance function* $k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, usually also known as *kernel*, such that $m(\boldsymbol{x}) = E\{f(\boldsymbol{x})\}$ and $k(\boldsymbol{x}, \boldsymbol{x}') = E\{(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))^T\}$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. A covariance function is valid for a GP prior specification if and only if it is symmetric and positive definite, that is, if $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$ and $\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) > 0$ for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. The associated correlation function is $\rho(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') / \sqrt{k(\boldsymbol{x}, \boldsymbol{x}) k(\boldsymbol{x}', \boldsymbol{x}')}$.

Without loss of generality, $m(\boldsymbol{x})$ is usually assumed to be the zero function. Instead, the choice of $k(\boldsymbol{x}, \boldsymbol{x}')$ is crucial as it essentially determines the behavior of the GP prior. Many competitors have been proposed during the years and the attention is usually restricted on *stationary covariance functions* being those expressible as functions of $\boldsymbol{x} - \boldsymbol{x}'$. GPs employing stationary covariance functions are referred

to as *stationary GPs*. Moreover, *isotropic covariance functions* are stationary covariance functions expressible as $k(\boldsymbol{x}, \boldsymbol{x}') = k(\|\boldsymbol{x} - \boldsymbol{x}'\|)$, such as the well-known *squared exponential covariance function*:

$$k_{\text{S.E.}}(\boldsymbol{x}, \boldsymbol{x}'; \tau^2, \ell^2) = k_{\text{S.E.}}(r; \tau^2, \ell^2) = \tau^2 \exp\left\{-\frac{r^2}{2\ell^2}\right\}, \quad r = \|\boldsymbol{x} - \boldsymbol{x}'\|. \tag{4.2}$$

It is parametrized by a signal variance (amplitude) parameter $\tau^2 > 0$ and a length-scale parameter $\ell^2 > 0$ which determines the oscillation frequency of the GP prior. The associated correlation function is itself isotropic (and stationary as well):

$$\rho_{\text{S.E.}}(\boldsymbol{x}, \boldsymbol{x}'; \ell^2) = \rho_{\text{S.E.}}(r; \ell^2) = \exp\left\{-\frac{r^2}{2\ell^2}\right\}, \quad r = \|\boldsymbol{x} - \boldsymbol{x}'\|. \tag{4.3}$$

The major attractiveness of GPs is that they can be interpreted as a collection of random variables, any finite number of which have a *consistent* joint Gaussian distribution. This means that, for any observed dataset $\{(\boldsymbol{x}_i, y_i) : 1 \leq i \leq n\}$ such that $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, $\boldsymbol{X} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$, $\boldsymbol{f}_{\boldsymbol{X}} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T$, $\boldsymbol{m}_{\boldsymbol{X}} = (m(\boldsymbol{x}_1), \ldots, m(\boldsymbol{x}_n))^T$ and $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} = (k(\boldsymbol{x}_i, \boldsymbol{x}_{i'}))_{1 \leq i,i' \leq n}$, then the GPR model admits the following intuitive *finite-sample representation*:

$$\boldsymbol{y}|\boldsymbol{f}_{\boldsymbol{X}} \sim \mathrm{N}_n(\boldsymbol{f}_{\boldsymbol{X}}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{f}_{\boldsymbol{X}} \sim \mathrm{N}_n(\boldsymbol{m}_{\boldsymbol{X}}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}).$$

Notice here and after we denote with proper subscripts the set of predictors over which the vectors and matrices are computed. Moreover, marginalizing out $\boldsymbol{f}_{\boldsymbol{X}}$ from the model specification yields $\boldsymbol{y} \sim \mathrm{N}_n(\boldsymbol{m}_{\boldsymbol{X}}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I})$.

As for any regression problem, the main use of GPR models is for predicting $\boldsymbol{f}_{\boldsymbol{X}^*} = (f(\boldsymbol{x}_1^*), \ldots, f(\boldsymbol{x}_{n^*}^*))^T$ over a set of n^* new predictors $\boldsymbol{X}^* = (\boldsymbol{x}_1^{*,T}, \ldots, \boldsymbol{x}_{n^*}^{*,T})^T$, exploiting the knowledge about the unknown regression function supplied by the observed data. Given the Bayesian nature of GPR model, this means obtaining the posterior predictive distribution of $\boldsymbol{f}_{\boldsymbol{X}^*}$ given the observed response vector $\boldsymbol{y}$. Notice the joint density function for $\boldsymbol{y}$ and $\boldsymbol{f}_{\boldsymbol{X}^*}$ is given by

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_{\boldsymbol{X}^*} \end{bmatrix} \sim \mathrm{N}_{n+n^*}\left(\begin{bmatrix} \boldsymbol{m}_{\boldsymbol{X}} \\ \boldsymbol{m}_{\boldsymbol{X}^*} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^*} \\ \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^*} & \boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*} \end{bmatrix}\right),$$

where again $\boldsymbol{m}_{\boldsymbol{X}^*} = (m(\boldsymbol{x}_1^*), \ldots, m(\boldsymbol{x}_{n^*}^*))^T$ and $\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*} = (k(\boldsymbol{x}_i^*, \boldsymbol{x}_{i'}^*))_{1 \leq i,i' \leq n^*}$, whereas $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^*} = (k(\boldsymbol{x}_i, \boldsymbol{x}_{i'}^*))_{1 \leq i \leq n, 1 \leq i' \leq n^*}$ and $\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}} = \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^*}^T$. The *posterior predictive distribution* follows immediately by standard conditioning formulas for multivariate

Gaussian distributions:

$$\begin{aligned} \boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y} \sim \mathrm{N}_{n^*}\Big(&\boldsymbol{m}_{\boldsymbol{X}^*} + \boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}}(\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{X}}), \\ &\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*} - \boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}}(\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^*}\Big). \end{aligned} \tag{4.4}$$

If the interest is that of predicting the unobserved response $\boldsymbol{y}_{\boldsymbol{X}^*} = (y_1^*, \ldots, y_{n^*}^*)^T$ instead of $\boldsymbol{f}_{\boldsymbol{X}^*}$, the same reasoning applies replacing $\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*}$ with $\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*} + \sigma^2\boldsymbol{I}$ and (4.4) turns into:

$$\begin{aligned} \boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y} \sim \mathrm{N}_{n^*}\Big(&\boldsymbol{m}_{\boldsymbol{X}^*} + \boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}}(\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{X}}), \\ &\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*} + \sigma^2\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}}(\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^*}\Big). \end{aligned} \tag{4.5}$$

Both expressions require the inversion of a matrix of size $n \times n$, which is an $\mathcal{O}(n^3)$ Cholesky operation unless $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$ has a special structure that can be exploited. Matrix inversion becomes more unstable due to the propagation of errors arising from finite machine precision, this problem being even more acute if $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$ is rank-deficient. Moreover, storing $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$ (and $\boldsymbol{K}_{\boldsymbol{X}^*,\boldsymbol{X}^*}$ as well) requires $\mathcal{O}(n^2)$ ($\mathcal{O}(n^{*2})$) dynamic memory. Once the inverse matrix has been computed, evaluation of $E(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y})$ is an $\mathcal{O}(n)$ operation and evaluation of $\mathrm{Var}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y})$ is $\mathcal{O}(n^2)$. The same holds for $E(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y})$ and $\mathrm{Var}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y})$. Therefore, a simple exact implementation of (4.4) and (4.5) can be handled efficiently with at most a few thousand observations. Many different proposals have been developed during the years to yield efficient computations of GPR models in large-data settings, see e.g. Smola and Bartlett (2001), Csató and Opper (2002), Quiñonero Candela and Rasmussen (2005), Chapter 8 of Rasmussen and Williams (2006), Banerjee *et al.* (2013) and Heaton *et al.* (2019) for exhaustive reviews.

Regardless of the mean and covariance functions adopted for (4.1), they are both usually defined in terms of proper parameters, which need to be estimated appropriately for making predictions with either (4.4) or (4.5). This is the case, for example, of the squared exponential covariance function (4.2) which is parametrized by τ^2 and ℓ^2. Moreover, the measurement errors variance σ^2 is generally not known.

The usual approach for jointly estimating the generic parameter vector $\boldsymbol{\theta}$ and σ^2 is by performing maximum-likelihood (usually called *type-II maximum likelihood* to highlight that we are doing maximum likelihood within a nonparametric Bayesian model) over the model marginal log-likelihood:

$$\log \mathfrak{p}(\boldsymbol{y}; \sigma^2, \boldsymbol{\theta}) = -\frac{n}{2}\log|\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I}| - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{X}})^T(\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2)^{-1}(\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{X}}).$$

Nevertheless, the presence of multiple modes can make the estimation process prone to overfitting, especially when the number of parameters is not moderate. Furthermore, weakly identified parameters can manifest in flat ridges for the marginal log-likelihood surface, making the optimization procedure extremely sensitive to the starting values as shown by Warnes and Ripley (1987). Overall, maximum likelihood estimates are subject to high variability and underestimate prediction uncertainty (Lalchand and Rasmussen, 2020).

4.2.2 Fully-Bayesian Inference

Taking a fully-Bayesian inferential approach to GPR requires specifying a prior distribution for $\boldsymbol{\theta}$ and σ^2 instead of treating them as unknown quantities to be estimated via maximum-likelihood. Therefore, the above formulas must be adequately adapted to exploit the fact that each quantity is conditionally dependent on the model parameters.

In particular, we specify $\boldsymbol{y}|\boldsymbol{f}_{\boldsymbol{X}}, \sigma^2 \sim \mathrm{N}_n(\boldsymbol{f}_{\boldsymbol{X}}, \sigma^2\boldsymbol{I})$ and $\boldsymbol{f}_{\boldsymbol{X}}|\boldsymbol{\theta} \sim \mathrm{N}_n(\boldsymbol{m}_{\boldsymbol{X}}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}})$, from which $\boldsymbol{y}|\sigma^2, \boldsymbol{\theta} \sim \mathrm{N}_n(\boldsymbol{m}_{\boldsymbol{X}}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})$ after marginalizing out $\boldsymbol{f}_{\boldsymbol{X}}$. Given $\mathfrak{p}(\sigma^2, \boldsymbol{\theta})$ the generic joint prior density function for $\boldsymbol{\theta}$ and σ^2, the posterior density function for the GPR model parameter is $\mathfrak{p}(\sigma^2, \boldsymbol{\theta}|\boldsymbol{y}) \propto \mathfrak{p}(\boldsymbol{y}|\sigma^2, \boldsymbol{\theta})\mathfrak{p}(\sigma^2, \boldsymbol{\theta})$. In general it does not admit an explicit expression, requiring efficient MCMC sampling schemes or variational approximation procedures to be implemented.

Regarding the predictive aspect of GPR, a fully-Bayesian inferential approach requires integrating out $\boldsymbol{\theta}$ and σ^2 from the posterior predictive expressions given above, employing the model posterior distribution. Hence the posterior predictive distribution for $\boldsymbol{f}_{\boldsymbol{X}^*}$ now becomes:

$$\mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}) = \int \mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \sigma^2, \boldsymbol{\theta})\mathfrak{p}(\sigma^2, \boldsymbol{\theta}|\boldsymbol{y}) \, \mathrm{d}\sigma^2 \, \mathrm{d}\boldsymbol{\theta}, \tag{4.6}$$

where $\mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \sigma^2, \boldsymbol{\theta})$ is the density function of (4.4) and, equivalently, the posterior predictive distribution for $\boldsymbol{y}_{\boldsymbol{X}^*}$ now becomes:

$$\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}) = \int \mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}, \sigma^2, \boldsymbol{\theta})\mathfrak{p}(\sigma^2, \boldsymbol{\theta}|\boldsymbol{y}) \, \mathrm{d}\sigma^2 \, \mathrm{d}\boldsymbol{\theta}, \tag{4.7}$$

where $\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}, \sigma^2, \boldsymbol{\theta})$ is the density function of (4.5). In both cases the GPR posterior predictive distribution can be interpreted as a mixture of multivariate Gaussian distributions, with $\mathfrak{p}(\sigma^2, \boldsymbol{\theta}|\boldsymbol{y})$ acting as mixing density. If variational approximations are employed, $\mathfrak{p}(\sigma^2, \boldsymbol{\theta}|\boldsymbol{y})$ is replaced by its optimal approximating density function

$\mathfrak{q}^*(\sigma^2, \boldsymbol{\theta})$. Although in general closed-form expressions do not exist, draws from either (4.6) or (4.7) can be obtained via Monte-Carlo, see e.g. the recent work of Lalchand and Rasmussen (2020).

4.3 Nonstationary Gaussian Process Regression

In absence of specific prior assumptions on the unknown regression function $f(\cdot)$, it is common use to specify a GP prior for it as in (4.1) selecting $k(\boldsymbol{x}, \boldsymbol{x}')$ to be stationary or, even more, isotropic. Reasons include their ability to accurately model a wide set of data patterns within accounting for a very limited number of parameters to be appropriately estimated (Stein, 1999).

Nonetheless, real data are often very far from the stationarity hypothesis: if stationary GPs are employed for data fitting and prediction in these scenarios, many complications may arise. For example, the estimation algorithm (whether a type-II maximum-likelihood or a Bayesian inferential approach is taken) could not converge successfully, or the predictions may highlight intrinsic narrowness in accurately grasping typical nonstationary changes in regime. Moreover, especially in spatial data settings, it is typical to employ isotropic covariance functions for performing GPR on data exhibiting inhomogeneous smoothness, i.e., exhibiting different degrees of smoothness in different subregions of $\mathcal{X}$. When the true latent regression function possesses heterogeneous behaviors or data indicates that stationary GP priors could not be an adequate choice, it is sensible to depart from the stationary hypothesis and consider nonstationary GP modeling strategies instead.

Many different approaches for nonstationarity GPR modeling have been proposed, and most of them encompass nonstationary covariance function specifications for the associated GP prior. Examples include vertical scaling (MacKay, 1998), input space warping (Sampson and Guttorp, 1992; Schmidt and O'Hagan, 2003), Bayesian treed GPs (Gramacy and Lee, 2008), composite GPs (Ba and Joseph, 2012) and convolution methods (e.g. Higdon *et al.*, 1999). Along this chapter, we focus on this latter attractive approach.

4.3.1 The Paciorek-Schervish Covariance Function

Convolution methods define a nonstationary covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{X}} K_{\boldsymbol{x}}(\boldsymbol{u}) K_{\boldsymbol{x}'}(\boldsymbol{u}) \, \mathrm{d}\boldsymbol{u}, \quad \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u} \in \mathcal{X} \subseteq \mathbb{R}^d,$$

for a *kernel function* $K_{\boldsymbol{x}}(\cdot)$ centered in the generic predictor $\boldsymbol{x}$. Paciorek (2003) and Paciorek and Schervish (2006) showed this covariance function is symmetric and positive definite for spatially-varying kernel functions of any form. Moreover, if $K_{\boldsymbol{x}}(\boldsymbol{u})$ is selected as being the probability density function of a d-variate $\mathrm{N}(\boldsymbol{u}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ random vector evaluated in $\boldsymbol{x}$, they derived a simple closed-form expression:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^T(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'})^{-1}(\boldsymbol{x} - \boldsymbol{x}')\right\}.$$

Absorbing the necessary constants into the matrices in the quadratic form and dividing by $\sqrt{k(\boldsymbol{x}, \boldsymbol{x})k(\boldsymbol{x}', \boldsymbol{x}')}$, the following nonstationary correlation function arise:

$$\rho(\boldsymbol{x}, \boldsymbol{x}') = \frac{2^{d/2}|\boldsymbol{\Sigma}_{\boldsymbol{x}}|^{1/4}|\boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/4}}{|\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/2}} \exp(-Q_{\boldsymbol{x},\boldsymbol{x}'}/2), \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$$
$$\text{with} \quad Q_{\boldsymbol{x},\boldsymbol{x}'} = (\boldsymbol{x} - \boldsymbol{x}')^T \left(\frac{\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'}}{2}\right)^{-1} (\boldsymbol{x} - \boldsymbol{x}').$$

The above correlation function has the form of an anisotropic squared exponential correlation function (that is, (4.3) with $\|\boldsymbol{x} - \boldsymbol{x}'\|^2/\ell^2$ replaced by the Mahalanobis distance $(\boldsymbol{x} - \boldsymbol{x}')^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{x}')$, for a pre-specified $\boldsymbol{\Sigma} \in \mathrm{S}_+^d$), but with the spatially-constant covariance matrix $\boldsymbol{\Sigma}$ replaced by the mean of two predictor-specific covariance matrices: $(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'})/2$. Hence, it can be written concisely as:

$$\rho(\boldsymbol{x}, \boldsymbol{x}') = \frac{2^{d/2}|\boldsymbol{\Sigma}_{\boldsymbol{x}}|^{1/4}|\boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/4}}{|\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/2}} \rho_{\text{S.E.}}\left(\sqrt{Q_{\boldsymbol{x},\boldsymbol{x}'}}; 1\right). \tag{4.8}$$

Notice if $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \boldsymbol{\Sigma}_{\boldsymbol{x}'}$ for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, then (4.8) reduces into the anisotropic – but stationary – squared exponential correlation function. Otherwise, the evolution of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}'}$ produces a nonstationary correlation function whose smoothness depends on both $\boldsymbol{x}$ and $\boldsymbol{x}'$.

Theorem 1 of Paciorek and Schervish (2006) extends (4.8) into a class of nonstationary correlation functions, later on called *Paciorek-Schervish correlation functions*, having generic expression:

$$\rho_{\text{P.S.}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{2^{d/2}|\boldsymbol{\Sigma}_{\boldsymbol{x}}|^{1/4}|\boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/4}}{|\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/2}} \rho\left(\sqrt{Q_{\boldsymbol{x},\boldsymbol{x}'}}\right) \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}, \tag{4.9}$$

where $\rho(r)$ is an isotropic correlation function with unit length-scale parameter. Similarly, *Paciorek-Schervish covariance functions* have generic expression:

$$k_{\text{P.S.}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{2^{d/2}|\boldsymbol{\Sigma}_{\boldsymbol{x}}|^{1/4}|\boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/4}}{|\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'}|^{1/2}} k\left(\sqrt{Q_{\boldsymbol{x},\boldsymbol{x}'}}\right) \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \tag{4.10}$$

where $k(r)$ is an isotropic covariance function with unit length-scale parameter. In both cases, $Q_{\boldsymbol{x},\boldsymbol{x}'} = (\boldsymbol{x} - \boldsymbol{x}')^T\{(\boldsymbol{\Sigma}_{\boldsymbol{x}} + \boldsymbol{\Sigma}_{\boldsymbol{x}'})/2\}^{-1}(\boldsymbol{x} - \boldsymbol{x}')$. Although not directly accounted in this chapter, Risser and Calder (2015) provided a slight generalization of the Paciorek-Schervish correlation function including spatially-varying standard deviations $\sigma_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}'}$ to be pre-multiplied to (4.9), hence adding further flexibility.

Regardless of the stationary covariance (correlation) function over which the Paciorek-Schervish covariance (correlation) function is built, the main challenge is to adequately specify the kernel matrices $\boldsymbol{\Sigma}_{\boldsymbol{x}}$, for any $\boldsymbol{x} \in \mathcal{X}$. In fact, their evolution determines how quickly the covariance structure changes over $\mathcal{X}$ and the degree to which the model adapts to predictor-dependent smoothness in the unknown regression function. In practice, since they are defined over an infinite-dimensional space, they must be parametrized in some way such that implementation is feasible: some widely-adopted techniques have been recently reviewed by Risser and Turek (2020).

One common strategy is that of forcing $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ to be locally isotropic, i.e., specifying $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \Sigma(\boldsymbol{x})\boldsymbol{I}_d$ for a proper $\Sigma(\boldsymbol{x}) : \mathcal{X} \longrightarrow \mathbb{R}^+$. Gibbs (1997) proposed a similar matrix diagonal specification, $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \text{diag}(\ell_1^2(\boldsymbol{x}), \ldots, \ell_d^2(\boldsymbol{x}))$ for a proper set of $\ell_i : \mathcal{X} \longrightarrow \mathbb{R}^+$ functions, $1 \leq i \leq d$, giving rise to the nonstationary *Gibbs covariance function*:

$$k_{\text{GIBBS}}(\boldsymbol{x}, \boldsymbol{x}') = \prod_{i=1}^{d} \left(\frac{2\ell_i(\boldsymbol{x})\ell_i(\boldsymbol{x}')}{\ell_i^2(\boldsymbol{x}) + \ell_i^2(\boldsymbol{x}')}\right)^{1/2} \exp\left\{-\frac{(x_i - x_i')^2}{\ell_i^2(\boldsymbol{x}) + \ell_i^2(\boldsymbol{x}')}\right\}, \quad \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}. \tag{4.11}$$

Each function ℓ_i works as a predictor-dependent length-scale parameter, one for each dimension, which needs to be specified appropriately. One interesting proposal is due to Plagemann *et al.* (2008), in which they assumed $\ell_i = \ell(\cdot)$ for each $1 \leq i \leq d$ and modeled $\log \ell(\cdot)$ with an independent stationary GP prior. Similar latent GP specifications have been proposed by, e.g., Paciorek and Schervish (2006), Tolvanen *et al.* (2014), Heinonen *et al.* (2016), Roininen *et al.* (2019) and Monterrubio-Gómez *et al.* (2020).

Interestingly, Dunlop *et al.* (2018) used (4.10) to propose a hierarchical construction of *deep Gaussian processes* (Damianou and Lawrence, 2013; Duvenaud *et al.*, 2014), i.e., deep belief networks (Hinton *et al.*, 2006) based on recursive GP mappings. In short words, they expressed a deep GP with $L + 1$ layers as a sequence of functions

$\{u_l(\boldsymbol{x})\}_{l=0}^{L}$ such that:

$$\begin{aligned} u_l(\boldsymbol{x})|u_{l+1} &\sim \mathcal{GP}(m(\boldsymbol{x};u_{l+1}); k_{\text{P.S.}}(\boldsymbol{x},\boldsymbol{x}';u_{l+1})), \quad 0 \leq l \leq (L-1), \\ u_L &\sim \mathcal{GP}(m(\boldsymbol{x}); k(\boldsymbol{x},\boldsymbol{x}')), \end{aligned} \tag{4.12}$$

where u_L takes a GP prior specification with a properly chosen mean function $m(\boldsymbol{x})$ and an isotropic covariance function $k(\boldsymbol{x},\boldsymbol{x}')$. All the L remaining GPs are connected one within the preceding one by specifying a Paciorek-Schervish covariance function $k_{\text{P.S.}}(\boldsymbol{x},\boldsymbol{x}';u_{l+1})$ being built over the adopted $k(\boldsymbol{x},\boldsymbol{x}')$, and specifying

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = F(u_{l+1}(\boldsymbol{x}))\boldsymbol{I}_d, \quad \boldsymbol{x} \in \mathcal{X} \tag{4.13}$$

for a properly chosen locally-bounded $F : \mathbb{R} \longrightarrow \mathbb{R}_{\geq 0}$. Notation $k_{\text{P.S.}}(\boldsymbol{x},\boldsymbol{x}';u_{l+1})$ highlights the dependency of the Paciorek-Schervish covariance function for the lth GP prior from the $(l+1)$th GP prior via (4.13). Similarly, notation $m(\boldsymbol{x};u_{l+1})$ explicitly accounts for possible strategies to introduce further dependency from the $(l+1)$th GP through its mean function specification, e.g., following the approach of Park and Choi (2010).

This hierarchical construction has the following Markovian interpretation: a stationary GP prior for u_L works as a *generating seed* over which $L-1$ additional GP priors for $u_l|u_{l+1}$ are built sequentially, each conditioned on the nonstationary behavior of the preceding one through the Paciorek-Schervish covariance function having $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ specified as in (4.13). Lastly, the bottom GP prior for $u_0|u_1$ is the one embedding the nonstationary construction of interest for the unknown regression function. The same approach of Dunlop *et al.* (2018) has been recently successfully employed by Zhao *et al.* (2021) to develop a hierarchical construction of deep state-space GPs, i.e., deep GPs also embedding a temporal dimension.

4.3.2 Model Specification

In this chapter, we propose a nonstationary GPR model which follows from the deep GP construction of Dunlop *et al.* (2018), although with just two layers. Simulated data examples will show that even only one latent layer can be useful to adequately account for a wide set of nonstationary patterns. Our specification extends the GPR model presented in Section 4.2 by replacing the stationary GP prior (4.1) for the unknown regression function with the hierarchical construction of (4.12)

having $L = 1$, namely:

$$y = f(\boldsymbol{x}) + \epsilon, \quad \epsilon \sim \mathrm{N}(0, \sigma^2),$$
$$f(\boldsymbol{x})|u \sim \mathcal{GP}(m_f(\boldsymbol{x}; u), k_{\text{P.S.–S.E.}}(\boldsymbol{x}, \boldsymbol{x}'; u, \tau_f^2)),$$
$$u(\boldsymbol{x}) \sim \mathcal{GP}(m_u(\boldsymbol{x}), k_{\text{S.E.}}(\boldsymbol{x}, \boldsymbol{x}'; \tau_u^2, \ell_u^2)).$$

Here $k_{\text{S.E.}}(\boldsymbol{x}, \boldsymbol{x}'; \tau_u^2, \ell_u^2)$ is specified as in (4.2) for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. Without loss of generality, a generic deterministic mean function for the GP priors on both layers is assumed, not depending on any additional model hyperparameter. Interestingly, we adopt a Paciorek-Schervish covariance function (4.10) being built over the isotropic squared exponential covariance function with fixed $\ell^2 = 1$. Following Dunlop *et al.* (2018), we choose $F(z) = z^2$ as the function mapping the latent stationary GP prior into a length-scale component for $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ being specified as in (4.13). An explicit expression for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ is the following:

$$k_{\text{P.S.–S.E.}}(\boldsymbol{x}, \boldsymbol{x}'; u, \tau_f^2) = \tau_f^2 \left(\frac{2|u(\boldsymbol{x})||u(\boldsymbol{x}')|}{u^2(\boldsymbol{x}) + u^2(\boldsymbol{x}')} \right)^{d/2} \exp\left\{ -\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{u^2(\boldsymbol{x}) + u^2(\boldsymbol{x}')} \right\}. \tag{4.14}$$

Notice this specification reduces into a particular case of the Gibbs covariance function (4.11) after setting $\ell_i(\boldsymbol{x}) = \ell(\boldsymbol{x}) = |u(\boldsymbol{x})|$ for any $1 \leq i \leq d$ and multiplying by an amplitude parameter τ_f^2. The main conceptual difference is that $|u(\boldsymbol{x})|$ can potentially be equal to zero: Proposition 2 of Dunlop *et al.* (2018) ensures the covariance function (4.14) – and any Paciorek-Schervish covariance function with kernel matrix specified as in (4.13) – is still positive-definite. Our approach can be immediately extended to account for specifications of (4.10) being built over more general isotropic covariance functions than the squared exponential, see Table 4.1 of Rasmussen and Williams (2006) for possible alternative choices. In the following, we will often indicate $f(\boldsymbol{x})|u$ as the *bottom layer (level)* of our nonstationary GPR model, and $u(\boldsymbol{x})$ as its *top layer (level)* or, equivalently, the *latent stationary GP prior*.

Suppose we observed a dataset $\{(\boldsymbol{x}_i, y_i) : 1 \leq i \leq n\}$, $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, $y \in \mathbb{R}$ and define the following vectors and matrices:

$$\boldsymbol{y} = (y_1, \ldots, y_n)^T, \quad \boldsymbol{X} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T,$$
$$\boldsymbol{f}_{\boldsymbol{X}} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T, \quad \boldsymbol{u}_{\boldsymbol{X}} = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_n))^T,$$
$$\boldsymbol{m}_{f;\boldsymbol{X}} = (m_f(\boldsymbol{x}_1; u_1), \ldots, m_f(\boldsymbol{x}_n; u_n))^T, \quad \boldsymbol{m}_{u;\boldsymbol{X}} = (m_u(\boldsymbol{x}_1), \ldots, m_u(\boldsymbol{x}_n))^T,$$
$$\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}} = (k_{\text{P.S.–S.E.}}(\boldsymbol{x}_i, \boldsymbol{x}_{i'}; u_i, u_{i'}, \tau_f^2))_{1 \leq i,i' \leq n}, \quad \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}} = (k_{\text{S.E.}}(\boldsymbol{x}_i, \boldsymbol{x}_{i'}; \tau_u^2, \ell_u^2))_{1 \leq i,i' \leq n},$$

Notice we explicitly indicated the unknown $u_i = u(\boldsymbol{x}_i)$ and $u_{i'} = u(\boldsymbol{x}_{i'})$ upon which $\boldsymbol{\Sigma}_{\boldsymbol{x}_i}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}_{i'}}$ are built, respectively. Then our nonstationary GPR model admits the following convenient fully-Bayesian finite-sample representation:

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{f}_{\boldsymbol{X}}, \sigma^2 \sim \mathrm{N}_n(\boldsymbol{f}_{\boldsymbol{X}}, \sigma^2\boldsymbol{I}), \quad \boldsymbol{f}_{\boldsymbol{X}}|\boldsymbol{u}_{\boldsymbol{X}}, \tau_f^2 \sim \mathrm{N}_n(\boldsymbol{m}_{f;\boldsymbol{X}}, \boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}), \\ \boldsymbol{u}_{\boldsymbol{X}}|\tau_u^2, \ell_u^2 \sim \mathrm{N}_n(\boldsymbol{m}_{u;\boldsymbol{X}}, \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}), \\ \sigma^2 \sim \text{Inverse-Gamma}(\xi_{\sigma^2}, \lambda_{\sigma^2}), \\ \tau_f^2 \sim \text{Inverse-Gamma}(\xi_{\tau_f^2}, \lambda_{\tau_f^2}), \\ \tau_u^2 \sim \text{Inverse-Gamma}(\xi_{\tau_u^2}, \lambda_{\tau_u^2}), \quad \ell_u^2 \sim \text{Inverse-Gamma}(\xi_{\ell_u^2}, \lambda_{\ell_u^2}). \end{aligned} \tag{4.15}$$

A generic Inverse Gamma prior distribution is placed over all the variance and length-scale parameters, with positive hyperparameters $\xi_{\sigma^2}, \lambda_{\sigma^2}, \xi_{\tau_f^2}, \lambda_{\tau_f^2}, \xi_{\tau_u^2}, \lambda_{\tau_u^2}, \xi_{\ell_u^2}$ and $\lambda_{\ell_u^2}$. Subscripts f and u are associated to the mean function and covariance function parameters for the bottom layer and top layer, respectively. Although not made explicit by the notation, the top layer covariance matrix $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}$ depends upon τ_u^2 and ℓ_u^2, while the bottom layer covariance matrix $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$ depends upon τ_f^2 and the unknown finite-sample representation $\boldsymbol{u}_{\boldsymbol{X}}$ of $u(\boldsymbol{x})$.

The posterior predictive distribution for such a model can be obtained following similar arguments given in Section 4.2. What changes here is that the unknown vector $\boldsymbol{u}_{\boldsymbol{X}}$ is associated with $\boldsymbol{X}$ but we are instead interested in predicting over a set $\boldsymbol{X}^*$ of new predictors. Therefore, $\boldsymbol{u}_{\boldsymbol{X}^*}$ is required for computing $\boldsymbol{m}_{f;\boldsymbol{X}^*,\boldsymbol{X}^*}$ and $\boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}^*}$ on the bottom layer, and we predict it with the top-layer specification employing (4.4) after substituting $\boldsymbol{y}$ with $\boldsymbol{u}_{\boldsymbol{X}}$ and removing the $\sigma^2\boldsymbol{I}$ addendum from both the mean vector and covariance matrix expressions. Theoretical details and derivations for the posterior predictive distribution on model (4.15) are given in Appendix D.1.

4.3.3 A Visualization of the Nonstationary GPR Model when $d = 1$

We now provide a concrete example of the vast heterogeneity of nonstationary GP prior patterns that can be accommodated within model specification (4.15). A zero mean function on both layers is specified, i.e. $m_f(\boldsymbol{x}; u) = m_u(\boldsymbol{x}) = 0$ for any $\boldsymbol{x} \in \mathcal{X}$. We focus on the $d = 1$ case to better visually grasp their behaviors.

Figure 4.1 displays twenty independent data replicates of size $n = 250$ simulated from the nonstationary GPR model over $\mathcal{X} = [-\pi, \pi]$. Each of them is obtained fixing the true parameter values $\sigma^2 = 0.025$ and $\tau_f^2 = 1$, the former embedding

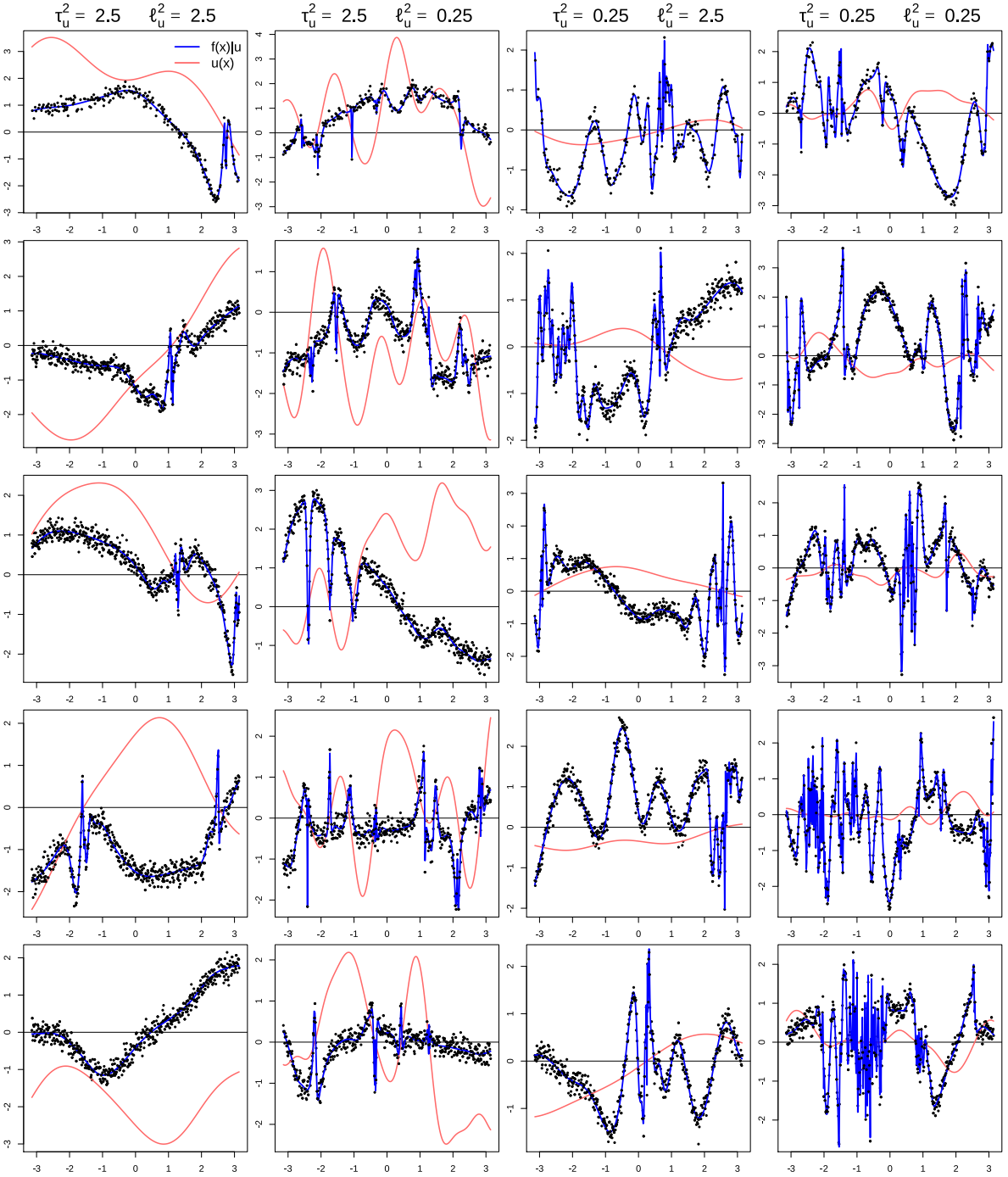

FIGURE 4.1: *Twenty univariate data replicates simulated from the nonstationary GPR model* (4.15) *having domain* $\mathcal{X} = [-\pi, \pi]$*, with fixed true parameter values* $\sigma^2 = 0.025$ *and* $\tau_f^2 = 1$*. Each column corresponds to a different combination of true parameter values* (τ_u^2, ℓ_u^2) *for the latent GP prior. For each of them, five possible realizations are displayed vertically.*

simulated data points being very close to realizations from $f(x)|u$. The predictor set $\boldsymbol{X} = (x_1, \ldots, x_{250})^T$ is composed by 250 equally-spaced points from $\mathcal{X}$, sorted in ascending order. Two different true values for the latent GP amplitude parameter τ_u^2 are considered: $\tau_u^2 = 2.5$ and $\tau_u^2 = 0.25$. The same true values have been considered for the latent GP length-scale parameter ℓ_u^2. Choice of $\ell_u^2 = 2.5$ leads to the specification of GP priors for $u(x)$ being very smooth, while $\ell_u^2 = 0.25$ leads to specifications exhibiting moderate oscillations around the x axis. Each column refers to one of the four different combinations of the true parameter values adopted for τ_u^2 and ℓ_u^2, as denoted at the top of each column, and vertically displays five independent data realizations. In each subfigure, the realization from the latent GP prior for $u(x)$ is represented with a solid red line, while that from the bottom GP prior for $f(x)|u$ with a solid blue line. Both are graphically displayed via linear interpolation of the vectors $\boldsymbol{u}_{\boldsymbol{X}}$ and $\boldsymbol{f}_{\boldsymbol{X}}|\boldsymbol{u}_{\boldsymbol{X}}$, respectively, simulated accordingly to (4.15).

The behavior of the stationary GP prior for $u(x)$, for each different combination of parameters τ_u^2 and ℓ_u^2, can be visually grasped having a look at the red line of each subfigure. Similarly, the behavior of the nonstationary GP prior realizations for $f(x)|u$ can be visually grasped by looking at the blue line in each subfigure. An intuition on how the latent GP prior for $u(x)$ influences the irregular behavior of the nonstationary GP prior for $f(x)|u$ is facilitated by choosing to plot realizations from both levels in the same subfigure. Therefore, it is immediate to notice that nonstationary patterns for the GP prior on $f(x)|u$ emerge in subintervals of $\mathcal{X}$ where the associated realization of the GP prior for $u(x)$ assumes values being very close to zero.

The reason is easily guessed by examining the expression for the nonstationary covariance function (4.14) when $d = 1$. For any $x \in \mathcal{X}$ and $x' \in \mathcal{X}$ in its closely-related neighborhood such that $u(x), u(x') \approx 0$, then a very small covariate-dependent length-scale contribution is embedded. Figure 4.2 shows the twenty heatmap representations for the covariance matrix $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$ involved in the finite-sample representation of the nonstationary GPR model (4.15), for each realization of the nonstationary GP prior on $f(x)|u$ displayed in Figure 4.1. A side-by-side comparison of the two figures allows to appreciate the nonstationary pattern introduced by (4.14), and how this reflects in the behavior for the realizations from the bottom layer. Sub-blocks of $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$ with behaviors different to the typical stationary block-diagonal bands have the effect of introducing nonstationary oscillations in the realizations from the GP prior over $f(x)|u$.

For clarity, consider the data replicate represented in the top-left subfigure of

Figure 4.1. In that case, the realization from the GP prior for $u(x)$ (red curve) makes the realization from the GP prior for the associated $f(x)|u(x)$ (blue curve) oscillating sensibly in $\mathcal{X}$, with a clear change in the regime of oscillation around $x \approx 3$, where $u(x) \approx 0$. An opposite behavior emerges from the nonstationary GP prior realization represented in the bottom-left subfigure of Figure 4.1. Here the realization from the GP prior for $u(x)$ is very smooth but oscillates very far from the x axis in all $\mathcal{X}$. Therefore, the realization from the GP prior for the associated $f(x)|u(x)$ preserves the same smooth behavior. Similar comments applies to all the other eighteen subfigures reported in Figure 4.1.

4.4 Approximate Bayesian Inference via GVA

Our main goal is to perform Bayesian inference on the nonstationary GPR model (4.15) and, as such, to obtain the posterior distribution over the model parameter vector. Standard MCMC sampling techniques cannot benefit from explicit conditionally-conjugate Gibbs sampling steps, although user-friendly interfaces such as Stan or Nimble can still be employed. For example, Risser and Turek (2020) recently used Nimble for performing Bayesian inference in high-dimensional nonstationary GPs.

The main obstacle introduced by the Paciorek-Schervish covariance function specification with $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ having expression (4.13) is that any finite representation of the nonstationary GPR model (4.15) yields an unknown latent vector $\boldsymbol{u}_{\boldsymbol{X}}$ whose dimension grows linearly with n. Therefore, the implementation of any MCMC algorithm becomes computationally prohibitive and does not scale well even with moderate data sample sizes. For this reason, we adopt GVA to obtain fast variational approximations to the posterior distribution sacrificing some degree of accuracy.

Let then $\boldsymbol{\lambda} = (\boldsymbol{u}_{\boldsymbol{X}}, \sigma^2, \tau_u^2, \ell_u^2, \tau_f^2)^T$ be the $(n+4)$-dimensional model parameter vector, $\boldsymbol{\lambda} \in \boldsymbol{\Lambda} \subseteq \mathbb{R}^n \times \mathbb{R}_+^4$, and define the following reparameterization map:

$$\begin{aligned} \psi \, : \boldsymbol{\Lambda} &\longrightarrow \boldsymbol{\Theta} \subseteq \mathbb{R}^{n+4} \\ \boldsymbol{\lambda} &\longrightarrow \boldsymbol{\theta} = \psi(\boldsymbol{\lambda}) = (\boldsymbol{u}_{\boldsymbol{X}}^T, \log \sigma^2, \log \tau_u^2, \log \ell_u^2, \log \tau_f^2)^T \end{aligned} \tag{4.16}$$

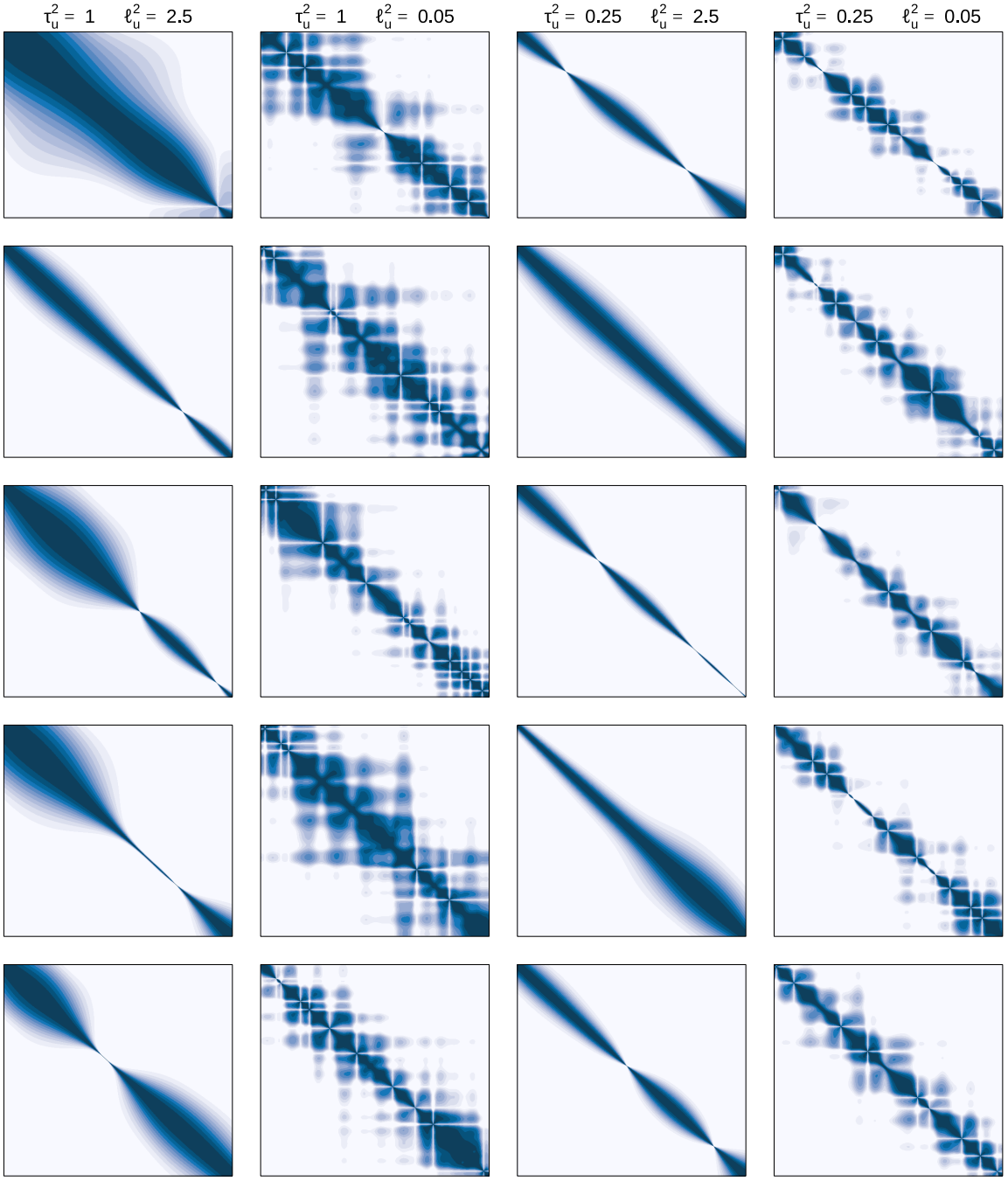

FIGURE 4.2: *Heatmap representation of the covariance matrix* $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$ *corresponding to the bottom layer random vector* $\boldsymbol{f}_{\boldsymbol{X}}|\boldsymbol{u}_{\boldsymbol{X}}$*, for each of the twenty simulated data replications showed in Figure 4.1. For each subfigure, the axis labels are not reported as they simply refer to the first and second dimension of a squared matrix.*

such that now each element of $\boldsymbol{\theta}$ takes values in the real line, as required by GVA. We approximate the model posterior distribution having density function

$$\begin{aligned} \mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \mathfrak{p}(\boldsymbol{y}|\boldsymbol{u}_{\boldsymbol{X}}, \log\sigma^2, \log\tau_f^2)\, \mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}}|\, \log\tau_u^2, \log\ell_u^2) \\ \times\, \mathfrak{p}(\log\sigma^2)\, \mathfrak{p}(\log\tau_u^2)\, \mathfrak{p}(\log\ell_u^2)\, \mathfrak{p}(\log\tau_f^2) \end{aligned} \tag{4.17}$$

with $\mathfrak{q}(\boldsymbol{\theta})$ having covariance matrix $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}$ parametrized with the latent factor approach of Ong *et al.* (2018), namely:

$$\mathfrak{q}(\boldsymbol{\theta}) \text{ is the } \mathrm{N}_{n+4}\left(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T + \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^2\right) \text{ density function .} \tag{4.18}$$

Here $\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} = (d_{\mathfrak{q}(\boldsymbol{\theta}),1}, \ldots, d_{\mathfrak{q}(\boldsymbol{\theta}),n+4})^T$ with $d_{\mathfrak{q}(\boldsymbol{\theta}),i} > 0$ for each $1 \leq i \leq n+4$, while $\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}$ is a full rank matrix of dimensions $(n+4) \times p$, for a pre-specified fixed number of latent factors $p \ll (n+4)$ and subject to the identifiability restriction $[\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}]_{ij} = 0$ for all $i < j$ (Geweke and Zhou, 1996). Moreover, $\mathfrak{p}(\log\sigma^2)$ in (4.17) is a notational artifact to express that it represents the probability density function of the transformed random variable $\log\sigma^2$, where $\sigma^2 \sim$ Inverse-Gamma$(\xi_{\sigma^2}, \lambda_{\sigma^2})$. Similar comment holds for $\mathfrak{p}(\log\tau_u^2)$, $\mathfrak{p}(\log\ell_u^2)$ and $\mathfrak{p}(\log\tau_f^2)$.

Notice the generic approximating density (4.18) embeds a covariance matrix $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}$ which is parametrized by $(n+4) \times (p+1)$ elements, a number which grows linearly in n and indeed is more suitable than typical Cholesky-factor parameterizations having a number of cells which, in our case, would grow quadratically in n. Ong *et al.* (2018) showed both with simulated and real data examples that a very limited number of latent factors p would assess optimal approximation performances.

An iterative scheme based on stochastic gradient ascent method for solving the associated optimization problem is well summarized in Section 3 of Ong *et al.* (2018) and adopted hereafter, following the same notation given in Section 1.4 of this PhD thesis. In short words, it is based upon the iterative refinement of the approximating density parameter vector $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} = (\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}^T, \mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})})^T, \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}^T)^T$ with the update expression (1.22) until convergence. The resulting $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^*$ then identifies the optimal approximate posterior distribution. An unbiased estimate for $\nabla_{\mathfrak{q}(\boldsymbol{\theta})} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))$ is given by (1.24) making use of the so-called reparameterization trick, after setting $\boldsymbol{\zeta} = (\boldsymbol{\varrho}^T, \boldsymbol{\omega}^T)^T$ with $\boldsymbol{\varrho}$ having dimension $(n+4) \times 1$ and $\boldsymbol{\omega}$ having dimension $p \times 1$, and noticing

$$\boldsymbol{\theta} = z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}) = \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} + (\boldsymbol{\omega}^T \otimes \boldsymbol{I})\mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}) + \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} \circ \boldsymbol{\varrho}.$$

Algorithm 4.1 *Gaussian Variational Approximation algorithm for determining the optimal natural parameter vector of the approximating density function* (4.18) *for approximate Bayesian inference of the nonstationary GPR model* (4.15).

Data Inputs: $\boldsymbol{y}$ $(n \times 1)$, $\boldsymbol{X}$ $(n \times d)$.

Select the number of latent factors p, such that $p \leq n + 4$.

Hyperparameter Inputs: ξ_{σ^2}, λ_{σ^2}, $\xi_{\tau_f^2}$, $\lambda_{\tau_f^2}$, $\xi_{\tau_u^2}$, $\lambda_{\tau_u^2}$, $\xi_{\ell_u^2}$ and $\lambda_{\ell_u^2}$, all > 0.

Initialize: $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}$, $\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}$ and $\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}$ with appropriate values.

$\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} \leftarrow (\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}^T, \text{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})})^T, \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}^T)^T$.

$E(\Delta^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) \leftarrow \mathbf{0}$, $E(g^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) \leftarrow \mathbf{0}$, $\varepsilon \leftarrow 10^{-6}$, $\nu \leftarrow 0.95$.

Cycle until convergence:

1. $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})} \longleftarrow \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T + \text{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^2$;
2. Sample $\boldsymbol{\varrho}$ from a $\text{N}_{n+4}(\mathbf{0}, \boldsymbol{I})$ distribution and $\boldsymbol{\omega}$ from a $\text{N}_p(\mathbf{0}, \boldsymbol{I})$ distribution;
3. Build the single draw from $\mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})})$ given $\boldsymbol{\varrho}$ and $\boldsymbol{\omega}$:
$$\tilde{\boldsymbol{\theta}} \longleftarrow \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} + \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}\boldsymbol{\omega} + \text{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})\boldsymbol{\varrho};$$
4. Compute the joint density gradient in $\tilde{\boldsymbol{\theta}}$:
$$\nabla_{\boldsymbol{\theta}} \log \mathfrak{p}(\boldsymbol{y}, \tilde{\boldsymbol{\theta}}) \longleftarrow \{\nabla_{\boldsymbol{\theta}} \log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})\}\big|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}};$$
5. Construct unbiased estimate of the gradient for the lower-bound:
$$\widehat{\nabla}_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})) \longleftarrow \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{\omega} \otimes \boldsymbol{I} \\ \text{diag}(\boldsymbol{\varrho}) \end{bmatrix} \left(\nabla_{\boldsymbol{\theta}} \log \mathfrak{p}(\boldsymbol{y}, \tilde{\boldsymbol{\theta}}) + (\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})})^{-1}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})})\right);$$
6. Update the adaptive learning rate $\boldsymbol{\rho}$ with the ADADELTA method:
$$\begin{aligned} E(g^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) &\longleftarrow \nu\, E(g^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) + (1-\nu)\left(\widehat{\nabla}_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))\right)^2; \\ \boldsymbol{\rho} &\longleftarrow \sqrt{E(\Delta^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) + \varepsilon} \Big/ \sqrt{E(g^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) + \varepsilon}\,; \\ E(\Delta^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) &\longleftarrow \nu\, E(\Delta^2_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}}) + (1-\nu)\left\{\boldsymbol{\rho} \circ \widehat{\nabla}_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))\right\}^2; \end{aligned}$$
7. Update $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}$:
$$\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} \longleftarrow \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} + \boldsymbol{\rho} \circ \widehat{\nabla}_{\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} \log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}));$$
8. Unpack $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}$ into $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}$, $\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}$, $\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}$:
$$\begin{aligned} \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} &\longleftarrow \text{ associated sub-vector of } \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}; \\ \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})} &\longleftarrow \text{vec}^{-1}_{(n+4)\times p}(\text{associated sub-vector of } \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}); \quad [\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}]_{ij} \longleftarrow 0 \text{ for all } i < j; \\ \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} &\longleftarrow \text{ associated sub-vector of } \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}. \end{aligned}$$

Outputs: $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}$, $\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}$, $\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}$.

A summarization of the GVA algorithm for our nonstationary GPR model with the above-mentioned parametrization for $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}$ is given in Algorithm 4.1, and follows from Algorithm 1 of Ong *et al.* (2018). Derivation and further implementation details are given in Appendix D.2. We adopt their same approach and use the ADADELTA method (Zeiler, 2012) for adaptively updating the learning rates of the stochastic gradient algorithm, with default tuning parameter choices $\varepsilon = 10^{-6}$ and $\nu = 0.95$.

Regarding the stopping criterion on which to establish the convergence of Algorithm 4.1, the usual strategy of monitoring the evolution of the lower-bound cannot be employed when solving GVA with stochastic gradient ascent methods. In fact, the update is stochastic and so the lower bound is not guaranteed to increase monotonically at each iteration. We adopt instead the approach described in Section 4.2 of Tan and Nott (2018): although computing the lower-bound explicitly requires evaluating the expectations of $\log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ and $\log \mathfrak{q}(\boldsymbol{\theta})$, it is straightforward to obtain an unbiased Monte-Carlo estimate at each iteration:

$$\widehat{\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))} = \log \mathfrak{p}(\boldsymbol{y}, \tilde{\boldsymbol{\theta}}) - \log \mathfrak{q}(\tilde{\boldsymbol{\theta}}; \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T + \operatorname{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^2).$$

We consider the average of these estimates over the past F iterations of the algorithm to minimize the variability, say $\overline{\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))}$. We compute it after every F iterations and keep a record of its maximum value attained thus far, say $\overline{\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))}_{\max}$. The algorithm is assumed to have reached convergence when $\overline{\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))}$ falls below $\overline{\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))}_{\max}$ more than M consecutive times, meaning that the lower bound estimates are just asymptotically bouncing around to its maxima.

Once the convergence of Algorithm 4.1 has been successfully reached, the optimal density function covariance matrix is $\boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})} = \boldsymbol{B}^*_{\mathfrak{q}(\boldsymbol{\theta})} \boldsymbol{B}^{*,T}_{\mathfrak{q}(\boldsymbol{\theta})} + \operatorname{diag}(\boldsymbol{d}^*_{\mathfrak{q}(\boldsymbol{\theta})})^2$ and

$$\mathfrak{q}^*(\boldsymbol{\theta}) \text{ is the } \mathrm{N}_{n+4}(\boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})}) \text{ density function .}$$

Consequently, the optimal approximating density function for the original model parameter vector $\boldsymbol{\lambda}$ can be obtained from $\mathfrak{q}^*(\boldsymbol{\lambda}) = \mathfrak{q}^*(\psi^{-1}(\boldsymbol{\lambda}))|\det \mathrm{J}_{\psi^{-1}}(\boldsymbol{\lambda})|$, where $\mathfrak{q}^*(\psi^{-1}(\boldsymbol{\lambda}))$ expresses $\mathfrak{q}^*(\boldsymbol{\theta})$ evaluated in $\boldsymbol{\theta} = \psi^{-1}(\boldsymbol{\lambda})$ and $\mathrm{J}_{\psi^{-1}}(\boldsymbol{\lambda})$ is the Jacobian matrix of the inverse reparameterization mapping (4.16). Nevertheless, in practical applications it is not necessary to find a closed-form expression for $\mathfrak{q}^*(\boldsymbol{\lambda})$: a generic draw $\tilde{\boldsymbol{\lambda}}$ from it can be simply obtained as $\tilde{\boldsymbol{\lambda}} = \psi^{-1}(\tilde{\boldsymbol{\theta}})$, $\tilde{\boldsymbol{\theta}}$ being sampled from a $\mathrm{N}_{n+4}(\boldsymbol{\mu}^*_{\mathfrak{q}(\boldsymbol{\theta})}, \boldsymbol{\Sigma}^*_{\mathfrak{q}(\boldsymbol{\theta})})$ distribution.

4.5 Numerical Investigations on Simulated Data

We now discuss the capabilities of our nonstationary GPR model (4.15) to obtain precise and affordable predictions for simulated datasets exhibiting clear nonstationarity patterns. Unlike the previous chapters, we have left aside the assessment of approximation accuracies and computational runtimes for obtaining GVA approximations compared to standard MCMC sampling procedures. Instead, we focused on investigating and evaluating the predictive ability of model (4.15) both visually and with typical error metrics, consistently with what is usually done in the machine-learning field for handling regression problems.

To do so, we simulated different univariate ($d = 1$) data replicates from model (4.15) fixing true parameter values $\sigma^2 = 0.025$, $\tau_f^2 = 1$, $\tau_u^2 = 1$ and $\ell_u^2 = 1.5$. Such a small value for σ^2 ensures generated observations to be very close to the true regression function of interest and mimics a real-data nonstationary scenario. Without loss of generality, all the other true parameter values have been chosen to avoid generation of data exhibiting clear stationary behaviors or *too irregular* nonstationary patterns requiring a huge number of training data to be effectively captured. A zero mean function on both layers is assumed, for which $\boldsymbol{m}_{f;\boldsymbol{X}} = \boldsymbol{m}_{u;\boldsymbol{X}} = \boldsymbol{0}$ in (4.15).

For each simulated dataset $\{\boldsymbol{X}_{\text{sim}}, \boldsymbol{y}_{\text{sim}}\}$, $\boldsymbol{X}_{\text{sim}}$ consists of $n_{\text{sim}} = 1000$ equally-spaced points from $\mathcal{X} = [-\pi, \pi]$ sorted in ascending order, and $\boldsymbol{y}_{\text{sim}}$ is composed by the associated response values generated accordingly to (4.15). We then sampled without repetition $n = 250$ pairs $(x_i, y_i)_{i=1}^n$ from the simulated dataset to compose the training dataset $\{\boldsymbol{X}, \boldsymbol{y}\}$, and $n^* = 500$ pairs $(x_i^*, y_i^*)_{i=1}^{n^*}$ to compose the testing dataset $\{\boldsymbol{X}^*, \boldsymbol{y}^*\}$ over which to evaluate the prediction accuracy. We specifically set a moderate value for n and doubled it for n^* to replicate a possible univariate real-data scenario in which few observed data points are available (but enough to capture the hidden nonstationary regression function pattern) and GPR is used to obtain predictions over a wide set of new predictors. We believe even better prediction results could be exploited by the proposed class of nonstatonary GPR models if enough training data are available such that they well describe the nonstationary patterns of interest.

For each training data replicate, Algorithm 4.1 has been employed to obtain GVA approximations for the model posterior distribution. Following advices from Ong *et al.* (2018) a very low number of latent factors parametrizing $\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\theta})}$ is selected, and numerical experiments always reached optimal convergence with $p = 5$. More ad-hoc procedures for adequately selecting p are firmly suggested if the obtained

approximations are not reliable. Weakly uninformative priors have been specified for the error variance and covariance functions parameters, setting $\xi_{\sigma^2} = 0.5$ and $\lambda_{\sigma^2} = 0.1$ for σ^2, while $\xi_{\tau_f^2} = \xi_{\tau_u^2} = \xi_{\ell_u^2} = \lambda_{\tau_f^2} = \lambda_{\tau_u^2} = \lambda_{\ell_u^2} = 0.5$ for τ_f^2, τ_u^2 and ℓ_u^2. Without loss of generality, $\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}$ have been initialized to be the zero vector, $\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}$ to have each cell on and below its diagonal equal to 0.001, and $\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}$ to be the vector full of ones. The convergence have always been assessed setting the default values $F = 500$ and $M = 3$ used by Tan and Nott (2018). Algorithm 4.1 have been entirely implemented and coded in R: nonetheless, evaluation of Step 4 requires automatic differentiation routines to compute $\nabla_{\boldsymbol{\theta}} \log \mathfrak{p}(\boldsymbol{y}, \tilde{\boldsymbol{\theta}})$ and we resorted to TensorFlow capabilities for efficient computations. See Appendix A.3 for further details.

Once the optimal GVA approximation has been determined, the model posterior predictive distribution $\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y})$ is obtained substituting $\mathfrak{q}^*(\boldsymbol{\lambda})$ to $\mathfrak{p}(\boldsymbol{\lambda}|\boldsymbol{y})$ in its expression derived in Appendix D.1. A generic sample from it represents a prediction $\tilde{\boldsymbol{y}}_{\boldsymbol{X}^*} = (\tilde{y}_1, \dots, \tilde{y}_{n^*})^T$ of the nonstationary GPR model (4.15) over the testing covariates $\boldsymbol{X}^*$. Therefore each $\tilde{y}_i$ can be compared with the *true* y_i^* associated to x_i^*, for each $1 \leq i \leq n^*$. Following a Monte-Carlo approach, we draw $B = 1000$ samples from $\mathfrak{q}^*(\boldsymbol{\lambda})$ and obtained a prediction $\tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}^{(b)}$ from $\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y})$, for each $1 \leq b \leq B$. We then compared each of them to the *true* $\boldsymbol{y}^*$ in testing dataset both with the popular *root mean square error (MSE)* and the *mean absolute error (MAE)* metrics, defined for a generic prediction $\tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}$ as

$$\text{RMSE}(\boldsymbol{y}^*, \tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}) = \sqrt{\frac{1}{n^*} \sum_{i=1}^{n^*} (y_i^* - \tilde{y}_i)^2} \quad \text{and} \quad \text{MAE}(\boldsymbol{y}^*, \tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}) = \frac{1}{n^*} \sum_{i=1}^{n^*} |y_i^* - \tilde{y}_i|,$$

respectively. We ended up with a distribution of B scores for both the RMSE and the MAE metrics, one for each simulated prediction.

To better grasp the advantages and possible limitations of our proposed nonstationary GPR model, we compared the obtained predictions to those produced by fitting its *stationary version* on each data replicate. Conceptually, this means removing the hidden layer inducing nonstationarity from model specification (4.15) and testing whether the stationary GP prior specified for its top layer can alone still handle comparable prediction performances. To allow for a fair comparison between the two predictions obtained, we computed GVA approximations with a modified

version of Algorithm 4.1 to handle the following stationary version of model (4.15):

$$\begin{gathered}\boldsymbol{y}|\boldsymbol{u}_{\boldsymbol{X}}, \sigma^2 \sim \mathrm{N}_n(\boldsymbol{u}_{\boldsymbol{X}}, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{u}_{\boldsymbol{X}}|\tau_u^2, \ell_u^2 \sim \mathrm{N}_n(\boldsymbol{m}_{u;\boldsymbol{X}}, \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}), \\ \sigma^2 \sim \text{Inverse-Gamma}(\xi_{\sigma^2}, \lambda_{\sigma^2}), \\ \tau_u^2 \sim \text{Inverse-Gamma}(\xi_{\tau_u^2}, \lambda_{\tau_u^2}), \quad \ell_u^2 \sim \text{Inverse-Gamma}(\xi_{\ell_u^2}, \lambda_{\ell_u^2}).\end{gathered} \tag{4.19}$$

Here $\boldsymbol{m}_{u;\boldsymbol{X}}$ and $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}$ exactly correspond to those specified in (4.15). The same hyperparameter values, initialization choices, and convergence criterion specified for the nonstationary GPR model have also been adopted here. Removing the hidden GP layer yields a generic approximating density function to be a multivariate Gaussian distribution of dimension 3. Therefore, we fitted a modified version of Algorithm 4.1 parametrizing $\boldsymbol{\Sigma}_{\mathrm{q}(\boldsymbol{\theta})}$ with the maximum possible number of latent factors: $p = 3$. Coherently, $B = 1000$ predictions from $\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y})$ having expression (4.5) have been obtained and compared to $\boldsymbol{y}^*$ using both the RMSE and MAE metrics.

Figures 4.3, 4.4, 4.5 and 4.6 displayed in the subsequent pages give a visual simmaruzation of the predictive results obtained for four different simulated data replicates. They have been carefully selected to encompass different nonstationary data patterns and therefore different prediction behaviors which allows us to highlight advantages and drawbacks of the GVA approximation strategy for our proposed nonstationary GPR model (4.15).

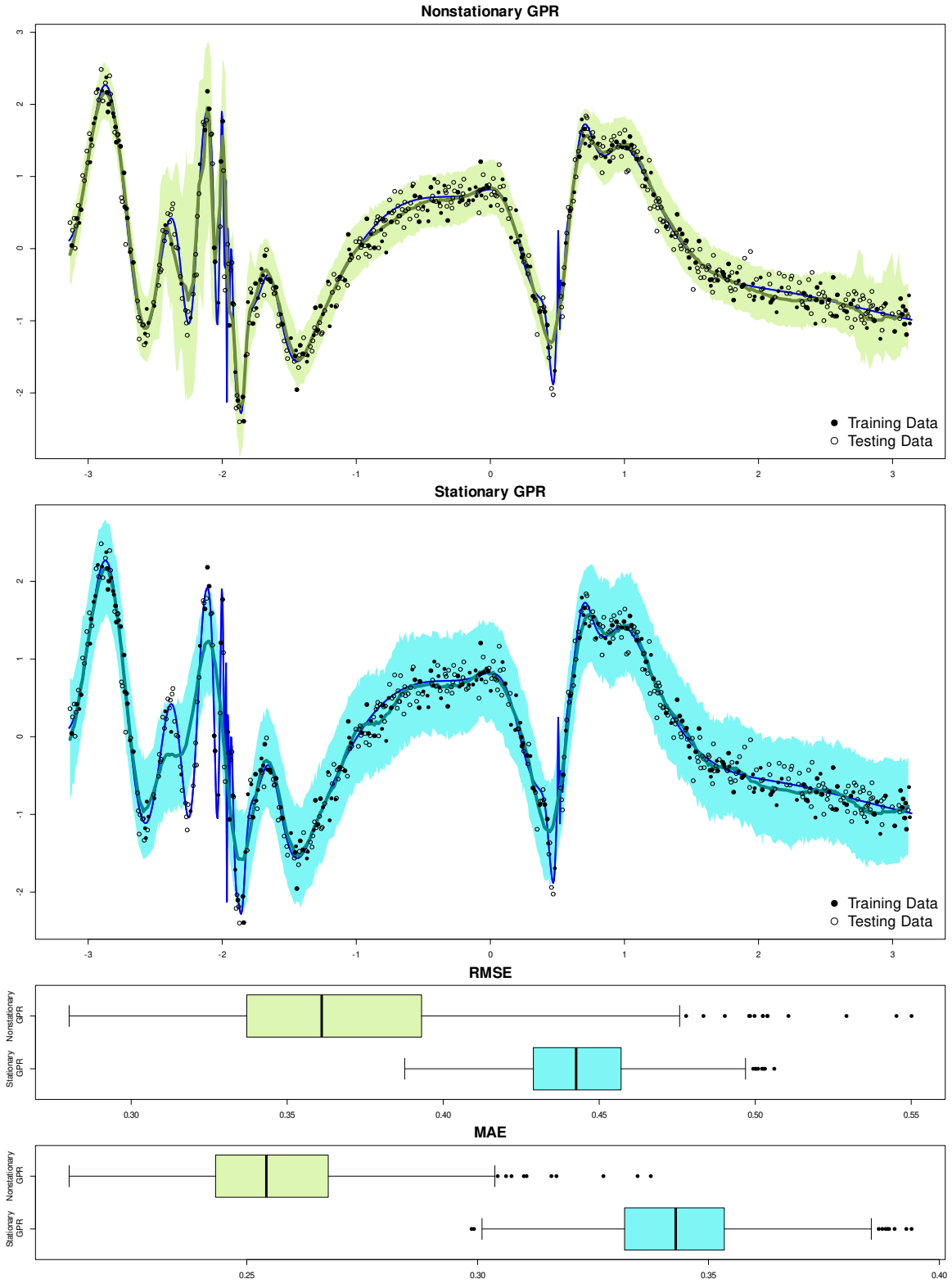

FIGURE 4.3: *Visual representation of the predictive behaviors for the nonstationary GPR model* (4.15) *and its stationary GPR counterpart* (4.19)*, both approximated via GVA. The first two subfigures display the posterior predictive bands and the median posterior predictive (thicker line) obtained by the two different models. The blue line represent the true regression function* $f(x)|u$*; the filled points represent the training data, the empty points the testing data. Both the blue line and the points are identical and repeated in the first two subfigures for visualization purposes. The third and fourth subfigures display the RMSE and MAE scores obtained from both models. This figure refers to the first simulated data replicate of interest to be commented.*

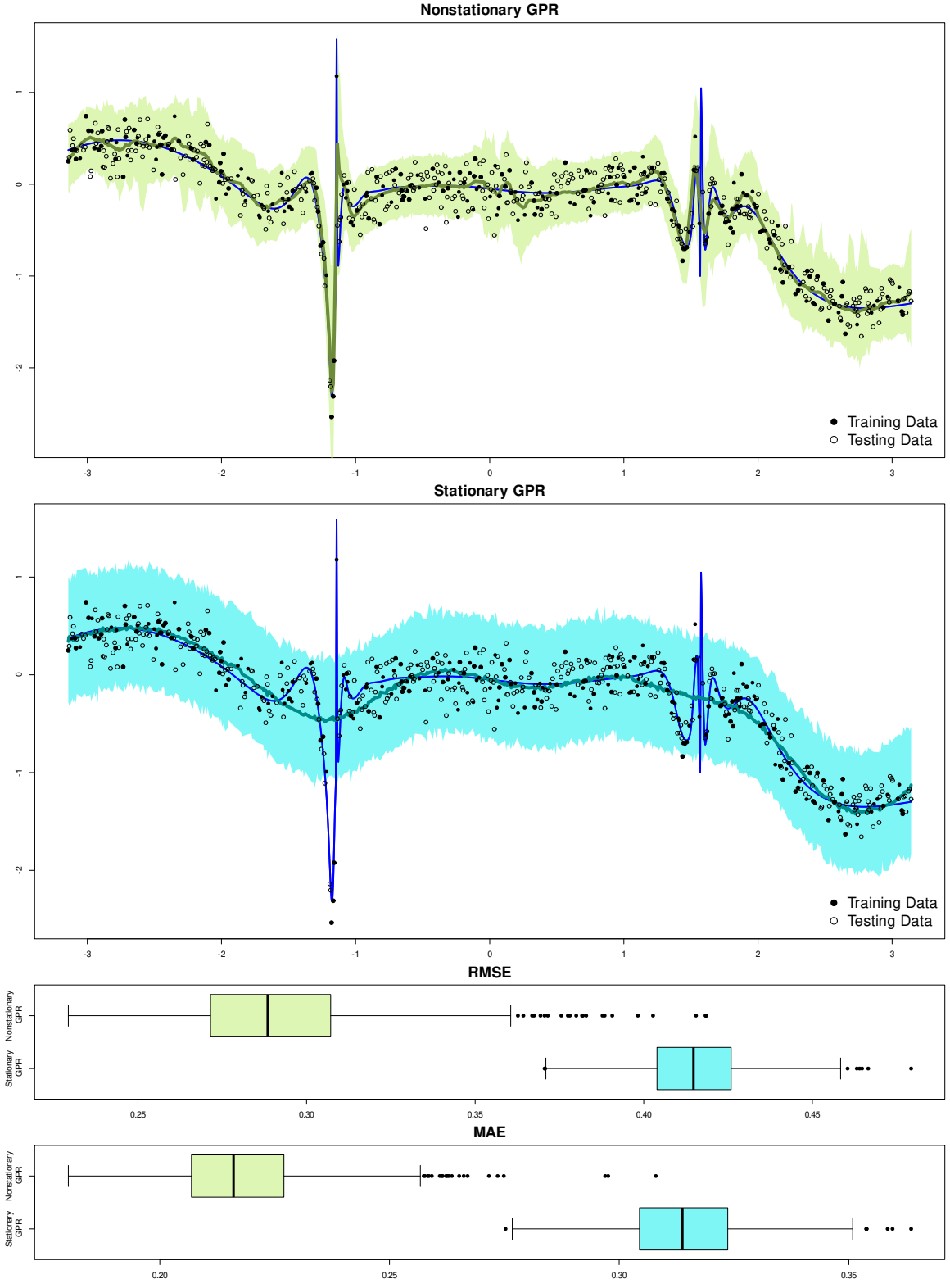

FIGURE 4.4: *Visual representation of the predictive behaviors for the nonstationary GPR model* (4.15) *and its stationary GPR counterpart* (4.19)*, both approximated via GVA. The first two subfigures display the posterior predictive bands and the median posterior predictive (thicker line) obtained by the two different models. The blue line represent the true regression function* $f(x)|u$*; the filled points represent the training data, the empty points the testing data. Both the blue line and the points are identical and repeated in the first two subfigures for visualization purposes. The third and fourth subfigures display the RMSE and MAE scores obtained from both models. This figure refers to the second simulated data replicate of interest to be commented.*

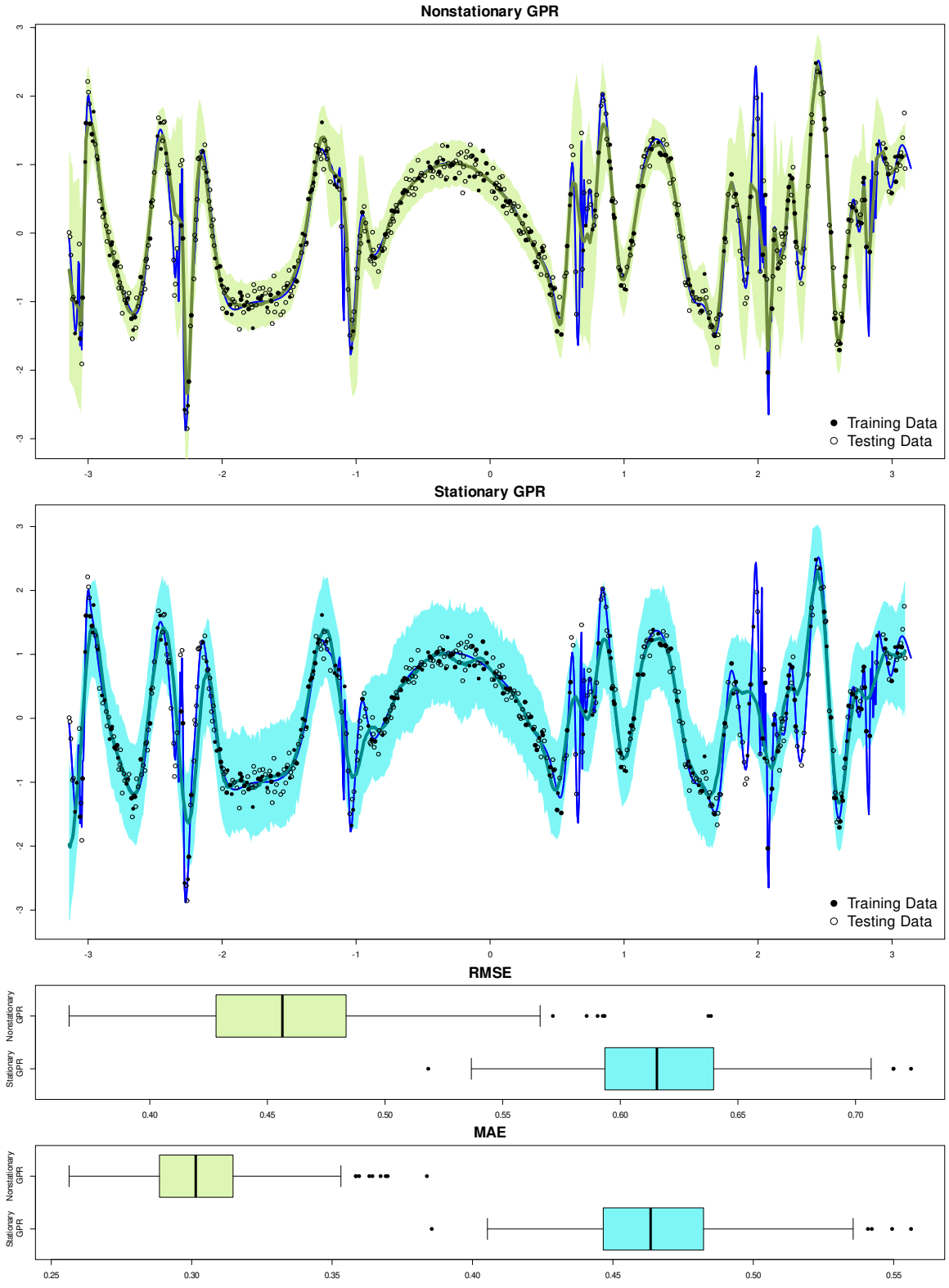

FIGURE 4.5: *Visual representation of the predictive behaviors for the nonstationary GPR model* (4.15) *and its stationary GPR counterpart* (4.19)*, both approximated via GVA. The first two subfigures display the posterior predictive bands and the median posterior predictive (thicker line) obtained by the two different models. The blue line represent the true regression function* $f(x)|u$*; the filled points represent the training data, the empty points the testing data. Both the blue line and the points are identical and repeated in the first two subfigures for visualization purposes. The third and fourth subfigures display the RMSE and MAE scores obtained from both models. This figure refers to the third simulated data replicate of interest to be commented.*

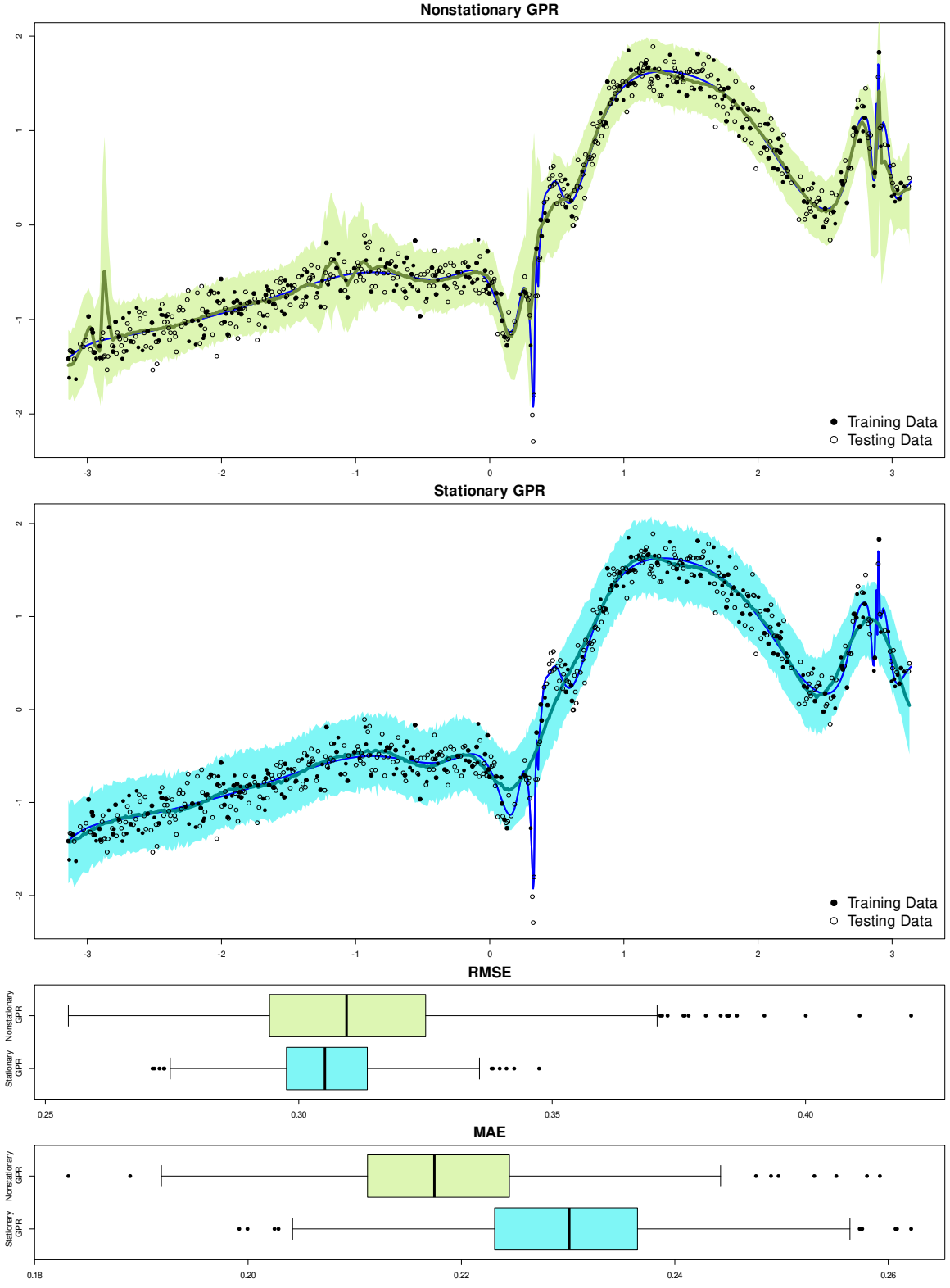

FIGURE 4.6: *Visual representation of the predictive behaviors for the nonstationary GPR model* (4.15) *and its stationary GPR counterpart* (4.19)*, both approximated via GVA. The first two subfigures display the posterior predictive bands and the median posterior predictive (thicker line) obtained by the two different models. The blue line represent the true regression function* $f(x)|u$*; the filled points represent the training data, the empty points the testing data. Both the blue line and the points are identical and repeated in the first two subfigures for visualization purposes. The third and fourth subfigures display the RMSE and MAE scores obtained from both models. This figure refers to the fourth simulated data replicate of interest to be commented.*

Each figure consists of four subfigures. Starting from top, the first two show the so-called *posterior predictive band* obtained from the nonstationary GPR model (4.15) (first subfigure) and from its stationary GPR counterpart (4.19) (second subfigure). Such bands have extremes obtained by linear interpolating the 95% HPD credible predictive intervals computed over the distribution of predictions $\tilde{y}_i^{(1)}, \ldots, \tilde{y}_i^{(B)}$ to x_i^*, for each $1 \leq i \leq n^*$ in the testing dataset. They allow to visually grasp within what extremes a generic prediction $\tilde{\boldsymbol{y}}$ from the associated model could be generated. In principle, a good Bayesian fit would have such a band containing all the observed data almost exactly, i.e., not being too narrow or too loose. Training observations are represented with filled points, while data-points belonging to the testing set with an empty point. The true unknown regression function $f(x)|u$ of interest from which data have been generated is displayed with a blue line. The thicker line results by linear interpolating the median of predictions $\tilde{y}_i^{(1)}, \ldots, \tilde{y}_i^{(B)}$ to x_i^*, for each $1 \leq i \leq n^*$, and reflects the posterior predictive median $\text{Median}_{\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y})}(\boldsymbol{y}_{\boldsymbol{X}^*})$ over $\boldsymbol{X}^*$. We expect such line superimposes the true regression function (blue line) if a good Bayesian fit has been obtained. The third subfigure displays with a boxplot the distribution of the RMSE scores for the B predictions obtained by the two GPR models considered. The last subfigure does the same for the MAE scores.

We start by commenting on the results from the first simulated data replicate of interest, as displayed in Figure 4.3. Data exhibit an evident change in the frequency of oscillations and, especially in the subinterval $[-2.5, -1.5]$, our nonstationary GPR model seems to better correctly predict the trend of the true regression function. In general, it also exhibits better prediction performances than the stationary GPR model, as confirmed by the distribution of both the RMSE and MAE scores. The reason is that the stationary GPR model accounts for a bigger variability in the predictions, as it is clear by comparing the amplitude for the prediction bands on the two subfigures, because we fit a stationary GPR model to nonstationary data. Our nonstationary GPR model still faces come difficulties in correctly grasping the amplitude of shocks on the true regression function, and we believe even better predictions could be obtained accounting for a bigger number of training and testing units in subintervals of $\mathcal{X}$ where such shocks arise. In general, posterior predictive bands for our nonstationary GPR model appear to be less smooth than those obtained for the stationary GPR model because of the hierarchical construction of the posterior predictive distribution.

Similar comments hold for all the other simulated data prediction results displayed. In particular, Figure 4.4 represents data following a stationary trend which

is interrupted by two major breaks around $x \approx -1$ and $x \approx 1.5$. Our nonstationary GPR model is able to capture that breaks and predictions correctly account for the behavior of the true regression function (blue line), to the best of their ability with such a limited number of testing units. On the contrary, the stationary GPR model completely ignores them. Figure 4.5 is an example of a clear nonstationary data pattern for which the stationary GPR model is moderately able to capture the true unknown regression function, although producing predictions fluctuating around it with a wider variability. RMSE and MAE summarize the more accurate predictions obtainable with the nonstationary GPR specification in both simulated data settings.

Finally, Figure 4.6 is an example in which predictions obtained by our nonstationary GPR model are slightly deteriorated than those obtained by fitting the stationary GPR model. RMSE and MAE scores confirm that our nonstationary GPR model provides no significant contribution in assessing more precise predictions than its stationary version. Nonetheless, it seems to better correctly capture the true regression function, especially in local subintervals in which it exhibits evident shocks. We believe predictions obtained by our nonstationary GPR model suffer both problems concerning numerical semi-definitive positiveness in the covariance matrices necessary to produce the predictions and a false convergence of the GVA algorithm towards an inaccurate approximation of the true posterior distribution.

4.6 Computational Bottlenecks and Possible Solutions

Despite its attractive prediction performances for fitting data exhibiting nonstationary patterns, model (4.15) is not exempt from computational bottlenecks involving both its specification and practical implementation of Algorithm 4.1.

As for any model involving GPs, data dimensionality is probably the most severe problem which makes practical implementations particularly onerous. The reason is that any finite representation of GPs, GPR models, and model (4.15) as well results being a multivariate Gaussian distribution of dimension n. When training data sample size has large dimensions, e.g., in spatial applications, even computing the model marginal likelihood is an $\mathcal{O}(n^3)$ operation which results in being extremely expensive both in terms of temporary memory consumption and computational runtime. Many proposals have been developed to yield efficient computations, as already mentioned in Section 4.2, and can be immediately extended to fit into our nonstationary GPR model specification.

A second major limitation of model (4.15) is that its finite representation embeds $\boldsymbol{u}_{\boldsymbol{X}}$, which is a latent vector of length n that cannot be marginalized out explicitly from the model specification unlike $\boldsymbol{f}_{\boldsymbol{X}}$. As such, it belongs to the global model parameter vector $\boldsymbol{\lambda}$ to all effects and makes the model posterior distribution to grow in size as $\mathcal{O}(n)$. Actually $\boldsymbol{u}_{\boldsymbol{X}}$ is a finite representation for the latent GP level $u(\boldsymbol{x})$ over the training covariate set $\boldsymbol{X}$, and we explicitly accounted for it on purpose because the determination of $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$ with the Paciorek-Schervish covariance function (4.14) requires $u_1 = u(\boldsymbol{x}_1), u_2 = u(\boldsymbol{x}_2), \ldots, u_n = u(\boldsymbol{x}_n)$. Nonetheless we can select $\tilde{n} \ll n$ predictors from $\boldsymbol{X}$, say $\tilde{\boldsymbol{X}}$, and consider the finite-sample representation of $u(\boldsymbol{x})$ in $\tilde{\boldsymbol{X}}$ instead of $\boldsymbol{X}$. Specifying

$$\begin{bmatrix} \boldsymbol{u}_{\boldsymbol{X}} \\ \boldsymbol{u}_{\tilde{\boldsymbol{X}}} \end{bmatrix} \Big| \tau_u^2, \ell_u^2 \sim \mathrm{N}_{n+\tilde{n}} \left(\begin{bmatrix} \boldsymbol{m}_{u;\boldsymbol{X}} \\ \boldsymbol{m}_{u;\tilde{\boldsymbol{X}}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}} & \boldsymbol{K}_{u;\boldsymbol{X},\tilde{\boldsymbol{X}}} \\ \boldsymbol{K}_{u;\tilde{\boldsymbol{X}},\boldsymbol{X}} & \boldsymbol{K}_{u;\tilde{\boldsymbol{X}},\tilde{\boldsymbol{X}}} \end{bmatrix} \right),$$

we can approximate the unknown $\boldsymbol{u}_{\boldsymbol{X}}$ required for computing $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$ as

$$\boldsymbol{u}_{\boldsymbol{X}} \approx E(\boldsymbol{u}_{\boldsymbol{X}} | \boldsymbol{u}_{\tilde{\boldsymbol{X}}}, \tau_u^2, \ell_u^2) = \boldsymbol{m}_{u;\boldsymbol{X}} + \boldsymbol{K}_{u;\boldsymbol{X};\tilde{\boldsymbol{X}}} \boldsymbol{K}_{u;\tilde{\boldsymbol{X}};\tilde{\boldsymbol{X}}}^{-1} (\boldsymbol{u}_{\tilde{\boldsymbol{X}}} - \boldsymbol{m}_{u;\tilde{\boldsymbol{X}}}),$$

after substituting $\boldsymbol{u}_{\tilde{\boldsymbol{X}}} | \tau_u^2, \ell_u^2 \sim \mathrm{N}_{\tilde{n}}(\boldsymbol{m}_{u;\tilde{\boldsymbol{X}}}, \boldsymbol{K}_{u;\tilde{\boldsymbol{X}},\tilde{\boldsymbol{X}}})$ to the specification for $\boldsymbol{u} | \tau_u^2, \ell_u^2$ in (4.15). Now the global model parameter vector $\boldsymbol{\lambda}$ has dimension $\tilde{n} + 4$ instead of $n + 4$. Algorithm 4.1 can still be employed for obtaining the GVA approximation of the model posterior distribution, with $\log \mathfrak{p}(\boldsymbol{y}; \boldsymbol{\theta})$ that has to be modified accordingly. We believe this method could be particularly effective and precise if $\tilde{\boldsymbol{X}}$ is accurately chosen such that it contains a subset of training predictors from $\boldsymbol{X}$ placed all over $\mathcal{X}$. Alternatively, Titsias and Lawrence (2010) variationally integrate out the latent variables using an expanded probability model in which the GP prior is augmented to include auxiliary inducing variables.

Lastly, we suggest entirely implementing Algorithm 4.1 in Python, which in our opinion better handles stochastic gradient ascent procedures than R and possesses a wide range of excellent libraries for handling GP features efficiently. Moreover, efficient usage of TensorFlow is guaranteed within a Python framework, where it has been developed and is currently officially maintained. We instead employed the Tensorflow API in R for doing fast automatic differentiation in Step 4 of Algorithm 4.1. It requires $\log \mathfrak{p}(\boldsymbol{y}; \boldsymbol{\theta})$ to be written in Tensorflow-like code and recursively computes it within R at each iteration: numerical experiments showed that this operation requires significant time for continuously transforming objects from R to Python and viceversa, and suggested us to switch into a full-Python implementation

instead. Such a task is currently under development, and we believe it could help to significatively speed up the overall runtime of Algorithm 4.1.

4.7 Concluding Remarks

Preliminary results obtained in this chapter motivate the attractiveness of studying variational approximation procedures for nonstationary modeling via GPR. They represent a fast alternative towards classical MCMC procedures and, to the best of our knowledge, have not yet been employed for nonstationary GPR modeling with Paciorek-Schervish covariance functions depending on fully-nonparametric latent GP priors. Most of the work present in literature is inspired by Section 3.2.2 of Paciorek and Schervish (2006), in which they specify latent GP priors in terms of their finite-basis function approximation (Kammann and Wand, 2003). Moreover, Dunlop *et al.* (2018) in their Section 5.1 mention variational approximations methods for deep GP constructions have not yet been studied in depth, further motivating our primitive work towards this direction. We are aware still lots of work has to be done, mostly regarding comparing our proposed methodology against alternative Bayesian approach for nonstationary regression diffuse in literature, and investigating its behavior in the $d = 2$ scenario. In particular, it will be fundamental to understand whether GVA approximations allows to still obtain satisfactory results both in terms of fitting and predictions, in comparison to classical Bayesian inferential approaches founded upon MCMC sampling. Such issues constitutes the keystones over which to expand this chapter.

The main focus for future work will also be on finding more adequate techniques for breaking down the need of resorting to the finite-sample representation of the latent $\boldsymbol{u}_X$ vector. This issue becomes central if a cascade of latent GP layers is added in (4.15), resorting to the characterization of deep GPs in terms of Paciorek-Schervish covariance functions proposed by Dunlop *et al.* (2018). Our nonstationary GPR model fits within the deep GP construction, although limiting the number of latent layers to be $L = 1$. While considering $L > 1$ adds further flexibility to the model specification and allows a broader set of nonstationary patterns to be fitted, it comes with nL additional parameters to be estimated. Hebbal *et al.* (2018) summarized alternative workarounds for dealing efficiently with deep GP expressed in terms of latent finite-sample GP representations.

Our proposed nonstationary GPR model immediately extends to account for alternative Paciorek-Schervish covariance functions in terms of more involved isotropic

covariance functions, e.g., using the Matern covariance function employed by Paciorek and Schervish (2006) in place of the squared exponential function. We preferred to *keep things simple* as not to overparametrize the model with useless parameters which may end up challenging to estimate adequately. Moreover, the latent GP layer can be connected within the bottom GP layer also through its mean function. Although not directly experimented in our work, few references in which this idea has been used are mentioned in the preceding pages.

Regarding the way we approximate the model posterior distribution, we believe the latent factor parameterization of Ong *et al.* (2018) best lends itself to do GVA in situations, as it is for our case, in which the model parameter vector dimension grows linearly with the sample size. This comes with the need of manually implementing the associated variational algorithm efficiently, with a particular emphasis on the automatic differentiation procedures required for determining the gradient of the log-joint posterior density function. An inexperienced user having few skills with such machine-learning topics would better resort to the ADVI procedure of Kucukelbir *et al.* (2017), which is completely automated and supported by Stan. It only requires the user to specify the model structure manually, and automatically sets up the approximating GVA procedure in a similar fashion of Algorithm 4.1, although taking care of itself of issues concerning efficient automatic differentiation. Nonetheless, it accounts for a Chokesly-factor decomposition of the approximating covariance matrix having a number of cells to be determined scaling up quadratically with the number of model parameters.

Conclusions

Discussion

Variational approximation methods have proved remarkably successful in many applicative fields and currently represent a popular alternative to MCMC sampling in Bayesian literature. Although variational Bayes approximations and variational approximation methods are frequently used as synonyms in the statistical field, the latter comprises a wider family of techniques – including VB itself – for facing the same optimization problem from different geometrical perspectives. Most of the variational approximation methods emerged from the machine learning field and are still mostly unexplored by the statistical community, because of the unattractive and often prohibitively difficult literature resources and the advanced programming skills required for efficient implementations of the underlying optimization schemes.

This PhD thesis studied some developments on the most popular variational approximation techniques, always keeping a Bayesian statistical viewpoint. Furthermore, we believe we have contributed to investigating currently open questions in the variational approximations literature in each chapter included in this manuscript. The only exception is Chapter 1, which is devoted to reviewing the literature and associated technicalities involved in variational approximation methods upon which the subsequent chapters are built.

In Chapter 2 we have developed explicit message-passing algorithms for finding variational approximations resulting from the minimization of a divergence measure alternative to the popular Kullback-Leibler divergence, recasting EP and VB approximations as opposite limiting cases and exploring the behavior of variational approximations conceptually lying in between those two. Although little significative improvements emerged from Power-EP approximations in comparison to VB and EP in all the statistical models considered, this allowed us to generalize – and

in some cases to improve significantly the implementations of – existing EP approximation algorithms proposed by Kim and Wand (2016) and Kim and Wand (2018).

The main research question addressed in Chapter 3 regards generalizing streamlined MFVB approximation techniques for multilevel models developed by Nolan *et al.* (2020) to accommodate global-local prior specifications over the fixed-effects vector. Besides, it seeks to experiment both with simulated and real data examples whether this more comprehensive set of priors effectively maintains the computational benefits carried out by their streamlined implementation over the naïve MFVB updating scheme. Global-local priors are particularly useful for implementing Bayesian selection procedures when a rich set of possible fixed-effects covariates is available, and it is of interest to discriminate between those effectively relevant in predicting the response variable and those that are not. We showed that an automated procedure proposed by Ray and Bhattacharya (2018) could be applied to the approximate posterior distributions obtained by our proposed algorithms for performing the selection. In contrast to alternative selection procedures proposed in Bayesian literature, it does not account for any hyperparameters tuning and, as far as our experiments are concerned, results in excellent fixed-effects selection performances.

Both Chapters 2 and 3 dedicate broad attention to studying the accuracy of the obtained optimal posterior approximations in contrast to the true posterior distribution of the model. This task is necessary to understand whether the obtained approximation is reliable or not but necessitates obtaining the true posterior distribution with MCMC sampling techniques. Therefore, it is not sustainable for practical implementations because the primary reason motivating variational approximations is to avoid recurring to MCMC sampling procedures.

With these considerations in mind, Chapter 4 follows a machine-learning approach and only focuses on investigating prediction performances obtained by GVA approximations over fully-Bayesian GPR models with nonstationary covariance functions specifications. We visually assessed the predictions obtained by this model over different simulated datasets and compared them to those obtained adopting a stationary GP specification with typical metrics employed in supervised regression learning. Keeping aside the accuracy investigation for the variational approximations obtained, we showed that our nonstationary GPR model better handles heterogeneous nonstationary patterns than standard stationary GPR models.

Future Directions of Research

Many future research directions can be envisaged for each of the works presented in this PhD thesis, and some possible extensions have been already pointed out in the *Closing Remarks* section of each chapter. Nonetheless, some common practical questions emerged from our work and we believe will guide research on variational approximation methods in the following years.

The first topic regards the way convergence is assessed for the optimization method employed to solve (1.1). The most theoretically-grounded strategy is monitoring the evolution of the lower-bound and stopping whenever its relative increment falls below a pre-specified threshold. Nonetheless, the lower-bound is not always guaranteed to increase monotonically for each approximating method. Moreover, its explicit expression could be challenging to be derived analytically and/or to compute without noticeably affecting the total runtime of each iteration. On the other hand, more practical alternative strategies employed in this PhD thesis include monitoring the relative absolute increment on the parameters for the approximating densities or letting the algorithm run over a number of iterations fixed in advance. Regardless of the strategy adopted, it is still not clear how to choose a coherent threshold such that the relative change obtained by performing one further iteration becomes negligible and, at the same time, the algorithm does not waste useless iterations while not obtaining any significant improvement in the overall approximation accuracy.

A second related topic is measuring the goodness of the proposed variational approximation and the associated optimal approximating densities. In principle, a suitable procedure would assess convergence in a very short time, scale with large model dimensions and sample sizes, and obtain optimal approximation accuracies to the true posterior distribution. As shown along with this PhD thesis, these benchmarks are usually not easy to measure genuinely and satisfy simultaneously. Moreover, the importance of each benchmark may vary from application to application and between different experimented approximating methods. We believe further research work has to be done towards understanding the accepted level of balance between a variational approximation that satisfies all the aforementioned benchmarks and one that sacrifices one or more of them in favor of a possibly more practical strategy for avoiding resorting to MCMC sampling strategies.

A third trending topic we believe will become highly predominant in the following years regards developing automated procedures for performing variational

approximations (Wingate and Weber, 2013), in a similar fashion to what popular probabilistic programming languages such as JAGS, Stan and Nimble already do for MCMC sampling. Deriving variational approximation procedures by hand often requires advanced mathematical skills and a lot of tedious algebraic computations to be handled manually or with numeric integrations methods; moreover, efficient implementations are necessary to enjoy the benefits provided by variational approximations. This may discourage an inexperienced user from experimenting the advantages of variational approximation methods in favor of relying to automated MCMC sampling strategies. We believe the ADVI procedure of Kucukelbir *et al.* (2017) represents the state of the art upon which to develop more flexible covariance matrix parameterization strategies such as that of Ong *et al.* (2018) experimented in Chapter 4. In addition, it should accommodate alternative approximations to GVA, e.g., exploiting the modularization of message-passing for MFVB and EP as already experimented by the Infer.NET software. With automated variational approximation procedures, the user only needs to specify the model structure and the type of approximation required, further contributing to disseminate the benefits of variational approximations into applicative fields in which they are still mostly unexplored.

Important theoretical research avenues for variational approximations have already been summarized in Chapter 1 as well as in popular review papers such as Blei *et al.* (2017) and Zhang *et al.* (2019), to which we refer for further insights.

Appendix A

A.1 A Primer on Vector Differential Calculus

Many statistical operations benefit from differential calculus, e.g., optimization of functions and calculation of information matrices. For multiparameter models, vector and matrix differential calculus circumvent element-wise univariate calculus and provide a concise way for expressing differential results on multivariate functions in terms of matrices and vectors.

Let f be a scalar-valued function with argument $\boldsymbol{x} \in \mathbb{R}^d$. The *derivative vector* of f, written $\nabla f(\boldsymbol{x})$, is the $d \times 1$ vector whose ith entry is

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i}.$$

The *Hessian matrix* of f, written $\mathsf{H}f(\boldsymbol{x})$, is the $d \times d$ matrix $\mathsf{H}f(\boldsymbol{x}) = \nabla(\nabla f(\boldsymbol{x}))^T$ whose (i,j)th entry is

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}.$$

As a shorthand notation, we usually denote with $\nabla_{\boldsymbol{x}}(\tilde{\boldsymbol{x}})$ the derivative vector of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, evaluated in $\tilde{\boldsymbol{x}}$, and equivalently $\mathsf{H}_{\boldsymbol{x}}(\tilde{\boldsymbol{x}})$ denotes the Hessian matrix of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, evaluated in $\tilde{\boldsymbol{x}}$.

We refer to Wand (2002) and Magnus and Neudecker (1988) for useful results concerning vector differential calculus.

A.2 Probability Distributions

We describe the probability distributions that are used or mentioned along with this PhD thesis. We distinguish them depending on whether or not they belong to the notable exponential family of distributions.

A.2.1 Exponential Families

The exponential family is a set of probability distributions that have been extensively studied in statistics due to their appealing properties. In particular, exponential family representations of probability density functions for distributions belonging to this family are commonly used in Bayesian statistics. They allow for efficient derivations of practical algorithms for approximate variational inference. For further theoretical insights, we refer to Barndorff-Nielsen (1978), Efron (1978) and Brown (1986).

We say that a $d \times 1$ random vector $\boldsymbol{X}$ has an *exponential family distribution* (meaning that its distribution belongs to the exponential family) if its probability density (or mass) function can be expressed as:

$$\mathfrak{p}(\boldsymbol{x};\boldsymbol{\eta}) = h(\boldsymbol{x}) \exp\left\{\boldsymbol{T}(\boldsymbol{x})^T\boldsymbol{\eta} - A(\boldsymbol{\eta})\right\}, \quad \boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{\eta} \in \mathbb{H}, \tag{A.1}$$

where $h(\boldsymbol{x}) \geq 0$ is the *base measure*, $\boldsymbol{T}(\boldsymbol{x})$ is the $d \times 1$ *sufficient statistic* vector, $\boldsymbol{\eta}$ is the $d \times 1$ *natural parameter* vector, $A(\boldsymbol{\eta})$ is the *log-partition function* and $\mathbb{H}$ is the space of allowable natural parameter values. The sufficient statistic and the log-partition function are linked by the two following useful results:

$$E(\boldsymbol{T}(\boldsymbol{X})) = \mathsf{D}_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T \quad \text{and} \quad \mathrm{Cov}(\boldsymbol{T}(\boldsymbol{X})) = \mathsf{H}_{\boldsymbol{\eta}} A(\boldsymbol{\eta}).$$

Many well-known distributions belong to the exponential family. We provide details and associated results focusing on those used in this PhD thesis.

A.2.1.1 Bernoulli Distribution

A discrete random variable X has a *Bernoulli* distribution with probability of success $\pi \in (0,1)$, written $X \sim \text{Bernoulli}(\pi)$, if its probability mass function is:

$$\mathfrak{p}(x;\pi) = \pi^x(1-\pi)^{1-x}, \quad x \in \{0,1\}.$$

The sufficient statistic and base measure are:

$$T(x) = x \quad \text{and} \quad h(x) = \mathbb{1}(x \in \{0,1\}).$$

The natural parameter and its inverse mapping are:

$$\eta = \log(\pi/(1-\pi) = \text{logit}(\pi) \quad \text{and} \quad \pi = e^{\eta}/(1+e^{\eta}) = \text{expit}(\eta),$$

and the log-partition function is:

$$A(\eta) = \log(1 + e^{\eta}).$$

Moreover,

$$E(T(X)) = 1/(1 + e^{-\eta}).$$

A.2.1.2 Univariate Normal (Gaussian) Distribution

A continuous random variable X has a *univariate Normal (Gaussian)* distribution with mean parameter $\mu \in \mathbb{R}$ and variance parameter $\sigma^2 > 0$, written $X \sim \mathrm{N}(\mu, \sigma^2)$, if its probability density function is:

$$\mathfrak{p}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \quad x \in \mathbb{R}.$$

When $\mu = 0$ and $\sigma = 1$, we refer to it as the univariate standard Normal (Gaussian) distribution, and indicate its probability density function with $\phi(x)$. The sufficient statistic and base measure are:

$$\boldsymbol{T}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix} \quad \text{and} \quad h(x) = (2\pi)^{-1/2}.$$

The natural parameter vector and its inverse mapping are:

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \mu/\sigma^2 \\ -1/(2\sigma^2) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mu \\ \sigma^2 \end{bmatrix} = \begin{bmatrix} -\eta_1/(2\eta_2) \\ -1/(2\eta_2) \end{bmatrix},$$

and the log-partition function is:

$$A(\boldsymbol{\eta}) = -\frac{1}{4}(\eta_1^2/\eta_2) - \frac{1}{2}\log(-2\eta_2).$$

Moreover,

$$E(\boldsymbol{T}(X)) = \begin{bmatrix} -\eta_1(2\eta_2) \\ (\eta_1^2 - 2\eta_2)/(4\eta_2^2) \end{bmatrix}.$$

A.2.1.3 Log-Normal Distribution

A continuous random variable X has a *Log-Normal* distribution with mean parameter $\mu \in \mathbb{R}$ and variance parameter $\sigma^2 > 0$, written $X \sim \text{Log-N}(\mu, \sigma^2)$, if its

probability density function is:

$$\mathfrak{p}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}x}\exp\left\{-\frac{(\log x-\mu)^2}{2\sigma^2}\right\}, \quad x>0.$$

If $X \sim$ Log-N(μ,σ^2), then $\log(X) \sim$ N(μ,σ^2). The sufficient statistic and base measure are:

$$\boldsymbol{T}(x) = \begin{bmatrix} \log x \\ \log^2 x \end{bmatrix} \quad \text{and} \quad h(x) = (2\pi)^{-1/2}.$$

The natural parameter vector, its inverse mapping and the log-partition function are identical to those of a univariate Normal distribution. Therefore,

$$E(\boldsymbol{T}(X)) = \begin{bmatrix} -\eta_1(2\eta_2) \\ (\eta_1^2 - 2\eta_2)/(4\eta_2^2) \end{bmatrix}.$$

A.2.1.4 Inverse Chi-Squared and Inverse Gamma Distributions

A continuous random variable X has an *Inverse Chi-Squared* distribution with shape parameter $\xi > 0$ and scale parameter $\lambda > 0$, written $X \sim$ Inverse-$\chi^2(\xi,\lambda)$, if its probability density function is:

$$\mathfrak{p}(x;\xi,\lambda) = \frac{(\lambda/2)^{\xi/2}}{\Gamma(\xi/2)}\, x^{-\xi/2-1}\exp\left\{-\frac{\lambda/2}{x}\right\}, \quad x>0.$$

A continuous random variable X has an *Inverse Gamma* distribution with shape parameter $\tilde{\xi} > 0$ and scale parameter $\tilde{\lambda} > 0$, written $X \sim$ Inverse-Gamma$(\tilde{\xi},\tilde{\lambda})$, if its probability density function is:

$$\mathfrak{p}(x;\tilde{\xi},\tilde{\lambda}) = \frac{\tilde{\lambda}^{\tilde{\xi}}}{\Gamma(\tilde{\xi})}\, x^{-\tilde{\xi}-1}\exp\left\{-\frac{\tilde{\lambda}}{x}\right\}, \quad x>0.$$

The Inverse Chi-Squared is not well-known, differently from the Inverse-Gamma distribution: just notice from their definitions that they both express the same probability distribution, just with a different parameterization. Explicitly,

$$X \sim \text{Inverse-}\chi^2(\xi,\lambda) \quad \text{if and only if} \quad X \sim \text{Inverse-Gamma}(\xi/2,\lambda/2)$$

or, equivalently,

$$X \sim \text{Inverse-Gamma}(\tilde{\xi},\tilde{\lambda}) \quad \text{if and only if} \quad X \sim \text{Inverse-}\chi^2(2\xi,2\lambda).$$

The sufficient statistic and base measure are:

$$\boldsymbol{T}(x) = \begin{bmatrix} \log x \\ 1/x \end{bmatrix} \quad \text{and} \quad h(x) = \mathbb{1}(x > 0).$$

The natural parameter vector and its inverse mapping are:

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} -(\xi + 2)/2 \\ -\lambda/2 \end{bmatrix} = \begin{bmatrix} -(\tilde{\xi} + 1) \\ -\tilde{\lambda} \end{bmatrix}, \quad \begin{bmatrix} \xi \\ \lambda \end{bmatrix} = \begin{bmatrix} -2 - 2\eta_1 \\ -2\eta_2 \end{bmatrix}$$

and

$$\begin{bmatrix} \tilde{\xi} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} -1 - \eta_1 \\ -\eta_2 \end{bmatrix}$$

and the log-partition function is:

$$A(\boldsymbol{\eta}) = (\eta_1 + 1)\log(-\eta_2) + \log\Gamma(-\eta_1 - 1).$$

Moreover,

$$E(\boldsymbol{T}(X)) = \begin{bmatrix} \log(-\eta_2) - \psi(-\eta_1 - 1) \\ (\eta_1 + 1)/\eta_2 \end{bmatrix}.$$

A.2.1.5 Gamma Distribution

A continuous random variable X has a *Gamma* distribution with shape parameter $\xi > 0$ and scale parameter $\lambda > 0$, written $X \sim \text{Gamma}(\xi, \lambda)$, if its probability density function is:

$$\mathfrak{p}(x; \xi, \lambda) = \frac{\lambda^{\xi}}{\Gamma(\xi)} x^{\xi - 1} \exp(-\lambda x), \quad x > 0.$$

The sufficient statistic and base measure are:

$$\boldsymbol{T}(x) = \begin{bmatrix} \log x \\ 1/x \end{bmatrix} \quad \text{and} \quad h(x) = \mathbb{1}(x > 0).$$

The natural parameter vector and its inverse mapping are:

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \xi - 1 \\ -\lambda \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \xi \\ \lambda \end{bmatrix} = \begin{bmatrix} 1 + \eta_1 \\ -\eta_2 \end{bmatrix}$$

and the log-partition function is:

$$A(\boldsymbol{\eta}) = -(\eta_1 + 1)\log(-\eta_2) + \log\Gamma(\eta_1 + 1).$$

Moreover,

$$E(\boldsymbol{T}(X)) = \begin{bmatrix} -\log(-\eta_2) + \psi(\eta_1 + 1) \\ -(\eta_1 + 1)/\eta_2 \end{bmatrix}.$$

A.2.1.6 Poisson Distribution

A discrete random variable X has a *Poisson* distribution with rate parameter $\lambda > 0$, written $X \sim \text{Poisson}(\lambda)$, if its probability mass function is:

$$\mathfrak{p}(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x \in \mathbb{N}.$$

The sufficient statistic and base measure are:

$$T(x) = x \quad \text{and} \quad h(x) = 1/(x!).$$

The natural parameter and its inverse mapping are:

$$\eta = \log\lambda \quad \text{and} \quad \lambda = e^{\eta}$$

and the log-partition function is:

$$A(\eta) = e^{\eta}.$$

Moreover,

$$E(T(X)) = e^{\eta}.$$

A.2.1.7 Inverse Gaussian Distribution

A continuous random variable X has an *Inverse Gaussian* distribution with mean parameter $\mu \in \mathbb{R}$ and precision parameter $\lambda > 0$, written $X \sim \text{Inverse-Gaussian}(\mu, \lambda)$, if its probability density function is:

$$\mathfrak{p}(x;\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi\, x^3}}\exp\left\{-\frac{\lambda(x-\mu)^2}{2\mu^2\, x}\right\}, \quad x > 0.$$

The sufficient statistic and base measure are:

$$\boldsymbol{T}(x) = \begin{bmatrix} x \\ 1/x \end{bmatrix} \quad \text{and} \quad h(x) = (2\pi\, x^3)^{-1/2}\mathbb{1}(x > 0).$$

The natural parameter vector and its inverse mapping are:

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} -\lambda/(2\mu^2) \\ -\lambda/2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mu \\ \lambda \end{bmatrix} = \begin{bmatrix} (\eta_2/\eta_1)^{1/2} \\ -2\eta_2 \end{bmatrix}$$

and the log-partition function is:

$$A(\boldsymbol{\eta}) = -2(\eta_1\eta_2)^{1/2} - \frac{1}{2}\log(-2\eta_2).$$

Moreover,

$$E(\boldsymbol{T}(X)) = \begin{bmatrix} (\eta_2/\eta_1)^{1/2} \\ (\eta_1/\eta_2)^{1/2} - 1/(2\eta_2) \end{bmatrix}.$$

A.2.1.8 Multivariate Normal (Gaussian) Distribution

A $d \times 1$ random vector $\boldsymbol{X}$ has a *multivariate Normal (Gaussian)* distribution with mean vector parameter $\boldsymbol{\mu} \in \mathbb{R}^d$ and symmetric positive definite $d \times d$ covariance matrix parameter $\boldsymbol{\Sigma}$, written $\boldsymbol{X} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if its probability density function is:

$$\mathfrak{p}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}, \quad \boldsymbol{x} \in \mathbb{R}^d.$$

When $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$, we refer to it as the multivariate standard Normal (Gaussian) distribution, and indicate its probability density function with $\boldsymbol{\phi}(\boldsymbol{x})$. The sufficient statistic and base measure are:

$$\boldsymbol{T}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{x} \\ \mathrm{vech}(\boldsymbol{x}\boldsymbol{x}^T) \end{bmatrix} \quad \text{and} \quad h(\boldsymbol{x}) = (2\pi)^{-d/2}.$$

The natural parameter vector and its inverse mapping are:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\mathrm{vech}(\boldsymbol{\Sigma}^{-1}) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\Sigma} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\{\mathrm{vech}^{-1}(\boldsymbol{\eta}_2)\}^{-1}\boldsymbol{\eta}_1 \\ -\frac{1}{2}\{\mathrm{vech}^{-1}(\boldsymbol{\eta}_2)\}^{-1} \end{bmatrix},$$

where $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ have dimensions $d \times 1$ and $d(d+1)/2 \times 1$, respectively, and the log-partition function is:

$$A(\boldsymbol{\eta}) = -\frac{1}{4}\boldsymbol{\eta}_1^T\{\text{vech}^{-1}(\boldsymbol{\eta}_2)\}^{-1}\boldsymbol{\eta}_1 - \frac{1}{2}\log|-2\text{vech}^{-1}(\boldsymbol{\eta}_2)|.$$

Moreover,

$$E(\boldsymbol{T}(\boldsymbol{X})) = \begin{bmatrix} -\frac{1}{2}\{\text{vech}^{-1}(\boldsymbol{\eta}_2)\}^{-1}\boldsymbol{\eta}_1 \\ \frac{1}{4}\text{vech}\left(\{\text{vech}^{-1}(\boldsymbol{\eta}_2)\}^{-1}\left[\boldsymbol{\eta}_1\boldsymbol{\eta}_1^T\{\text{vech}^{-1}(\boldsymbol{\eta}_2)\}^{-1} - 2\boldsymbol{I}\right]\right) \end{bmatrix}.$$

A.2.1.9 Inverse Wishart and Inverse G-Wishart Distributions

A $d \times d$ symmetric positive definite random matrix $\boldsymbol{X}$ has an *Inverse Wishart* distribution with $\kappa > d-1$ degrees of freedom and symmetric and positive definite $d \times d$ scale matrix parameter $\boldsymbol{\Lambda}$, written $\boldsymbol{X} \sim \text{Inverse-Wishart}(\kappa, \boldsymbol{\Lambda})$, if its probability density function is:

$$\mathfrak{p}(\boldsymbol{X}; \kappa, \boldsymbol{\Lambda}) = \frac{|\boldsymbol{\Lambda}|^{\kappa/2}}{2^{d\kappa/2}\pi^{d(d-1)/4}\prod_{j=1}^{d}\Gamma\left(\frac{\kappa+1-j}{2}\right)}|\boldsymbol{X}|^{-(\kappa+d+1)/2}\exp\left\{-\frac{1}{2}\text{tr}(\boldsymbol{\Lambda}\boldsymbol{X}^{-1})\right\}$$
$$\times\, \mathbb{1}(\boldsymbol{X} \text{ is symmetric and positive definite}).$$

When $d = 1$, it coincides with the Inverse Chi-Squared distribution. The sufficient statistic and base measure are:

$$\boldsymbol{T}(\boldsymbol{X}) = \begin{bmatrix} \log|\boldsymbol{X}| \\ \text{vech}(\boldsymbol{X}^{-1}) \end{bmatrix} \quad \text{and} \quad h(\boldsymbol{x}) = \frac{\mathbb{1}(\boldsymbol{X} \text{ is symmetric and positive definite})}{\pi^{d(d-1)/4}}.$$

The natural parameter vector and its inverse mapping are:

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} = \begin{bmatrix} -(\kappa+d+1)/2 \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Lambda}) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \kappa \\ \boldsymbol{\Lambda} \end{bmatrix} = \begin{bmatrix} -d-1-2\eta_1 \\ -2\text{vech}^{-1}(\boldsymbol{\eta}_2) \end{bmatrix},$$

where $\boldsymbol{\eta}_2$ has dimension $d(d+1)/2 \times 1$, and the log-partition function is:

$$A(\boldsymbol{\eta}) = \left(\eta_1 + \frac{d+1}{2}\right)\log|-\text{vech}^{-1}(\boldsymbol{\eta}_2)| + \sum_{j=1}^{d}\log\Gamma\left(-\eta_1 - \frac{d+j}{2}\right).$$

Moreover,

$$E(\boldsymbol{T}(\boldsymbol{X})) = \begin{bmatrix} \log|-\operatorname{vech}^{-1}(\boldsymbol{\eta}_2)| - \sum_{j=1}^{d} \psi\left(-\eta_1 - \frac{d+j}{2}\right) \\ \left(\eta_1 + \frac{d+1}{2}\right) \operatorname{vech}\left[\left(\operatorname{vech}^{-1}(\boldsymbol{\eta}_1)\right)^{-1}\right] \end{bmatrix}.$$

Now consider the extension of the Inverse Wishart distribution corresponding to the inverse of the $d \times d$ random matrix $\boldsymbol{X}$ having some off-diagonal entries forced to equal zero. Such structure can be represented using undirected graphs and, following the nomenclature of Atay-Kayis and Massam (2005), Letac and Massam (2007) and Uhler *et al.* (2018) is referred to as the Inverse G-Wishart distribution. Differently, Roverato (2000) used the term Hyper Inverse Wishart for the same family of distributions. Let then G be an undirected graph with d nodes labelled $1, \ldots, d$ and E be the edge set consisting of pairs of nodes that are connected by an edge. We say that the $d \times d$ matrix $\boldsymbol{M}$ *respects* G if:

$$\boldsymbol{M}_{ij} = 0 \quad \text{for all} \quad \{i, j\} \notin E.$$

Then the $d \times d$ symmetric positive definite random matrix $\boldsymbol{X}$ has an *Inverse G-Wishart* distribution with d-node undirected graph G, shape parameter $\xi > 0$ and symmetric positive definite $d \times d$ scale matrix parameter $\boldsymbol{\Lambda}$, written $\boldsymbol{X} \sim \text{Inverse-G-Wishart}(G, \xi, \boldsymbol{\Lambda})$, if its probability density function is:

$$\mathfrak{p}(\boldsymbol{X}; \xi, \boldsymbol{\Lambda}) \propto |\boldsymbol{X}|^{-(\xi+2)/2} \exp\left\{-\frac{1}{2}\operatorname{tr}(\boldsymbol{\Lambda}\boldsymbol{X}^{-1})\right\} \times \mathbb{1}(\boldsymbol{X} \text{ is symmetric and positive definite and } \boldsymbol{X}^{-1} \text{ respects } G).$$

The normalizing factor follows from formulae of Uhler *et al.* (2018), although it is pretty complicated for a general graph G. Two notable special cases arise, depending on the structure of G.

The $G = G_{\text{full}}$ and $\xi > 2(d-1)$ Case

If $G = G_{\text{full}} =$ totally connected d-node graph, symbolizing $\boldsymbol{X}^{-1}$ is a full matrix, then the Inverse G-Wishart distribution coincides with the ordinary Inverse Wishart distribution and $\boldsymbol{X} \sim \text{Inverse-G-Wishart}(G_{\text{full}}, \xi, \boldsymbol{\Lambda})$ if and only if $\boldsymbol{X} \sim \text{Inverse-Wishart}(\xi - d + 1, \boldsymbol{\Lambda})$. Then, its probability density function is expressible

as:

$$\mathfrak{p}(\boldsymbol{X};\xi,\boldsymbol{\Lambda}) = \frac{|\boldsymbol{\Lambda}|^{(\xi-d+1)/2}}{2^{d(\xi-d+1)/2}\pi^{d(d-1)/4}\prod_{j=1}^{d}\Gamma\left(\frac{\xi-d-j}{2}+1\right)}|\boldsymbol{X}|^{-(\xi+2)/2}\exp\left\{-\frac{1}{2}\text{tr}(\boldsymbol{\Lambda}\boldsymbol{X}^{-1})\right\}$$
$$\times\, \mathbb{1}(\boldsymbol{X} \text{ is symmetric and positive definite and } \boldsymbol{X}^{-1} \text{ respects } G),$$

and $\boldsymbol{X}$ effectively belongs to the exponential family. Its sufficient statistic, base measure, natural parameter vector, inverse mapping and log-partition function can be obtained from those of the Inverse Chi-Squared distribution, substituting $\kappa = \xi - d + 1$. Also, $E(\boldsymbol{X}^{-1}) = (\xi - d + 1)\boldsymbol{\Lambda}^{-1}$.

The $G = G_{\text{diag}}$ Case

Conversely, if $G = G_{\text{diag}} =$ totally disconnected d-node graph, symbolizing $\boldsymbol{X}^{-1}$ is a diagonal matrix and $E = \emptyset$, then the Inverse G-Wishart distribution reduces into a product of independent Inverse Chi-Squared distributions. This means $\boldsymbol{X} \sim$ Inverse-G-Wishart$(G_{\text{diag}}, \xi, \boldsymbol{\Lambda})$ if and only if $\boldsymbol{X} = \prod_{j=1}^{d} X_{jj}$, with $X_{jj} \overset{\text{ind}}{\sim}$ Inverse-$\chi^2(\xi, \Lambda_{jj})$ for all $1 \leq j \leq d$. In fact, its probability density function is:

$$\mathfrak{p}(\boldsymbol{X};\xi,\boldsymbol{\Lambda}) = \frac{|\boldsymbol{\Lambda}|^{\xi/2}}{2^{d\xi/2}\Gamma(\xi/2)^d}|\boldsymbol{X}|^{-(\xi+2)/2}\exp\left\{-\frac{1}{2}\text{tr}(\boldsymbol{\Lambda}\boldsymbol{X}^{-1})\right\}\prod_{j=1}^{d}\mathbb{1}(X_{jj}>0),$$

which can be expressed as:

$$\mathfrak{p}(\boldsymbol{X};\xi,\boldsymbol{\Lambda}) = \prod_{j=1}^{d}\left\{\frac{(\Lambda_{jj}/2)^{\xi/2}}{\Gamma(\xi/2)}X_{jj}^{-\xi/2-1}\exp\left(-\frac{1}{2}\frac{\Lambda_{jj}}{X_{jj}}\right)\mathbb{1}(X_{jj}>0)\right\}.$$

Then, $\boldsymbol{X}$ effectively belongs to the exponential family. Its sufficient statistic, base measure, natural parameter vector, inverse mapping and log-partition function can be obtained from that of the Inverse Chi-Squared distribution, substituting $\kappa = \xi - d + 1$. Also, $E(\boldsymbol{X}^{-1}) = \xi\boldsymbol{\Lambda}^{-1} = \xi\text{diag}(\Lambda_{11}^{-1}, \ldots, \Lambda_{dd}^{-1})$. See Maestrini and Wand (2021) for further insights.

A.2.2 Other Useful Distributions

Other distributions not belonging to the exponential family are used throughout this PhD thesis. We list them hereafter, together with associated results allowing for their algebraic tractability.

A.2.2.1 Uniform Distribution

A continuous random variable X has a *Uniform* distribution over the interval $[\nu_{\min}, \nu_{\max}]$, with $-\infty < \nu_{\min} < \nu_{\max} < \infty$, written $X \sim \text{Unif}[\nu_{\min}, \nu_{\max}]$, if its probability density function is:

$$\mathfrak{p}(x; \nu_{\min}, \nu_{\max}) = (\nu_{\max} - \nu_{\min})^{-1} \mathbb{1}(x \in [\nu_{\min}, \nu_{\max}]).$$

A.2.2.2 Half-Cauchy Distribution

A continuous random variable X has a *Half-Cauchy* distribution with scale parameter $s > 0$, written $X \sim \text{Half-Cauchy}(s)$, if its probability density function is:

$$\mathfrak{p}(x; s) = \frac{2}{\pi s\{1 + (x/s)^2\}}, \quad x > 0.$$

Proposition 1 in Armagan *et al.* (2011) showed that:

$$\text{if} \quad X^2|a \sim \text{Inverse-Gamma}(1/2, 1/a) \quad \text{and} \quad a \sim \text{Inverse-}\chi^2(1/2, 1/s^2),$$
$$\text{then} \quad X \sim \text{Half-Cauchy}(s),$$

or alternatively, using the Inverse-χ^2 parameterization:

$$\text{if} \quad X^2|a \sim \text{Inverse-}\chi^2(1, 1/a) \quad \text{and} \quad a \sim \text{Inverse-}\chi^2(1, 1/s^2),$$
$$\text{then} \quad X \sim \text{Half-Cauchy}(s).$$

A.2.2.3 Half-t Distribution

A continuous random variable X has a *Half-t* distribution with $\nu > 0$ degrees of freedom and scale parameter $s > 0$, written $X \sim \text{Half-}t(s, \nu)$, if its probability density function is:

$$\mathfrak{p}(x; \nu) = \frac{2\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma(\nu/2)s\{1 + (x/s)^2/\nu\}^{(\nu+1)/\nu}}, \quad x > 0.$$

If $\nu = 1$, then the Half-t distribution reduces into the Half-Cauchy distribution. Result 5 of Wand *et al.* (2011) showed that:

$$\text{if} \quad X^2|a \sim \text{Inverse-Gamma}(\nu/2, \nu/a) \quad \text{and} \quad a \sim \text{Inverse-}\chi^2(1/2, 1/s^2),$$
$$\text{then} \quad X \sim \text{Half-}t(s, \nu),$$

or alternatively, using the Inverse-χ^2 parameterization:

$$\text{if} \quad X^2|a \sim \text{Inverse-}\chi^2(\nu, 1/a) \quad \text{and} \quad a \sim \text{Inverse-}\chi^2(1, 1/(\nu s^2)),$$
$$\text{then} \quad X \sim \text{Half-}t(s, \nu).$$

A.2.2.4 Laplace Distribution

A continuous random variable X has a *Laplace* distribution (also known as *double-Exponential* or *two-tailed Exponential*) with mean parameter $\mu \in \mathbb{R}$ and scale parameter $\sigma > 0$, written $X \sim \text{Laplace}(\mu, \sigma)$, if its probability density function is:

$$\mathfrak{p}(x; \mu, \sigma) = \frac{1}{2\sigma} \exp\left(-\frac{|x - \mu|}{\sigma}\right), \quad x \in \mathbb{R}.$$

A useful result by Andrews and Mallows (1974) and West (1987) showed that:

$$\text{if} \quad X|\zeta \sim \text{N}(\mu, \sigma^2/\zeta) \quad \text{and} \quad \zeta \sim \text{Inverse-}\chi^2(2, 1),$$
$$\text{then} \quad X \sim \text{Laplace}(\mu, \sigma).$$

A.2.2.5 Horseshoe Distribution

A continuous random variable X has a *Horseshoe* distribution with mean parameter $\mu \in \mathbb{R}$ and scale parameter $\sigma > 0$, written $X \sim \text{Horseshoe}(\mu, \sigma)$, if its probability density function is:

$$\mathfrak{p}(x; \mu, \sigma) = (2\pi^3)^{-1/2} \frac{1}{\sigma} \exp\left\{\frac{(x - \mu)^2}{2\sigma^2}\right\} E_1\left\{\frac{(x - \mu)^2}{2\sigma^2}\right\}, \quad x \in \mathbb{R},$$

where $E_1(x) = \int_x^\infty t^{-1} e^{-t} \, \mathrm{d}t$ for $x \neq 0$ is the *exponential integral function of order 1*. Integrating Carvalho *et al.* (2010) with results from the Half-Cauchy distribution, it is possible to show that:

$$\text{if} \quad X|\zeta \sim \text{N}(\mu, \sigma^2/\zeta), \quad \zeta|a_\zeta \sim \text{Gamma}(1/2, a_\zeta) \quad \text{and} \quad a_\zeta \sim \text{Gamma}(1/2, 1),$$
$$\text{then} \quad X \sim \text{Horseshoe}(\mu, \sigma).$$

A.2.2.6 Normal-Exponential-Gamma Distribution

A continuous random variable X has a *Normal-Exponential-Gamma* distribution with mean parameter $\mu \in \mathbb{R}$, scale parameter $\sigma > 0$ and shape parameter $\lambda > 0$,

written $X \sim \text{NEG}(\mu, \sigma, \lambda)$, if its probability density function is:

$$\mathfrak{p}(x; \mu, \sigma, \lambda) = \pi^{-1/2} \lambda 2^{\lambda} \Gamma(\lambda + 1/2) \frac{1}{\sigma} \exp\left\{ \frac{(x-\mu)^2}{4\sigma^2} \right\} D_{-2\lambda-1}\left\{ \left| \frac{x-\mu}{\sigma} \right| \right\}, \quad x \in \mathbb{R},$$

where $D_{\nu}(x) = 2^{\nu/2+1/4} W_{\nu/2+1/4,-1/4}(x^2/2)/\sqrt{(}x)$ for $x > 0$ is the *parabolic cylinder function* of order $\nu \in \mathbb{R}$ and $W_{k,m}$ is a *confluent hypergeometric function* of order k and m, as defined by Whittaker and Watson (1996). A useful result by Griffin and Brown (2011) showed that:

$$\text{if} \quad X|\zeta \sim \text{N}(\mu, \sigma^2/\zeta), \quad \zeta|a_{\zeta} \sim \text{Inverse-}\chi^2(2, 2a_{\zeta}) \quad \text{and} \quad a_{\zeta} \sim \text{Gamma}(\lambda/2, 1),$$
$$\text{then} \quad X \sim \text{NEG}(\mu, \sigma, \lambda).$$

A.2.2.7 Huang-Wand Distribution

A $d \times d$ symmetric positive definite random matrix $\boldsymbol{X}$ has a *Huang-Wand* distribution with shape parameter $\nu > 0$ and scale parameters $s_1, \ldots, s_d > 0$, written $\boldsymbol{X} \sim \text{Huang-Wand}(\nu, \{s_1, \ldots, s_d\})$, if its probability density function is:

$$\mathfrak{p}(\boldsymbol{X}; \nu, \{s_1, \ldots, s_d\}) \propto |\boldsymbol{X}|^{-(\nu+2d)/2} \prod_{j=1}^{d} \{\nu \boldsymbol{X}_{jj}^{-1} + 1/s_j^2\}^{-(\nu+d)/2}\}$$
$$\times \mathbb{1}(\boldsymbol{X} \text{ is symmetric and positive definite}).$$

The name comes from the two authors who invented it and extensively studied its appealing properties in Huang and Wand (2013). As highlighted in Section 3.1 of Maestrini and Wand (2021), equation (2) of Huang and Wand (2013) with d rather than p and s_j rather than A_j for $1 \leq j \leq d$ allows to succinctly express it in terms of tractable Inverse G-Wishart distributions:

$$\text{if} \quad \boldsymbol{X}|\boldsymbol{A} \sim \text{Inverse-G-Wishart}(G_{\text{full}}, \nu + 2d - 2, \boldsymbol{A}^{-1})$$
$$\text{and} \quad \boldsymbol{A} \sim \text{Inverse-G-Wishart}(G_{\text{diag}}, 1, \{\nu \, \text{diag}(s_1^2, \ldots, s_d^2)\}^{-1}),$$
$$\text{then} \quad \boldsymbol{X} \sim \text{Huang-Wand}(\nu, \{s_1, \ldots, s_d\}).$$

As discussed in Huang and Wand (2013), the Huang-Wand distribution generalizes the Half-t distribution for random matrices. In fact, $(\boldsymbol{X}_{jj})^{1/2} \overset{\text{ind}}{\sim} \text{Half-}t(s_j, \nu)$ for each $1 \leq j \leq d$ and the choice $\nu = 2$ induces:

$$\frac{\sqrt{\boldsymbol{X}_{jj'}}}{\sqrt{\boldsymbol{X}_{jj} \, \boldsymbol{X}_{j'j'}}} \sim \text{Uniform}(-1, 1) \quad \text{for each } j \neq j'.$$

A.3 Hardware Setup, Programming Languages and Related Libraries

We briefly describe the hardware setup, programming languages, and associated libraries that have been used for writing this PhD thesis.

A.3.1 Hardware Setup

All the figures, simulation studies, applications, and related computational tasks presented in this PhD thesis have been executed into an R environment v4.0.3 and performed on a MacBook Pro laptop having a 1.4 gigahertz processor and 8 gigabytes of random access memory, running a macOS Big Sur v11.6 operating system. This manuscript has been written with LaTeX using texpad[1] v1.9.6 for macOS, and typesetted with pdfTeX[2] v3.14159265-2.6-1.40.21.

A.3.2 R

R (R Core Team, 2020) is a programming language and free software environment for statistical computing and graphics supported by the *R Core Team* and the *R Foundation for Statistical Computing*. It was created and developed by Ross Ihaka and Robert Gentleman during the 1990s, with its first stable version released in 2000. It is widely used among statisticians for developing statistical software and data analysis. The official R software is written primarily in C, Fortran and R itself and is freely available under the GNU General Public License. Although R has a command-line interface, several third-party graphical user interfaces are available, such as the popular RStudio (`https://www.rstudio.com`) and Jupyter (`https://jupyter.org`). Its popularity has increased rapidly over the last decades because it is open-source, it is available both for Windows, Mac and Linux operating systems, has an expanding set of freely available libraries and packages to extend its capabilities and has an extensive support network with numerous online and freely available documentation. Additional information can be found on the official website: `https://www.r-project.org`.

[1] see `https://www.texpad.com`.

[2] see `https://www.tug.org/applications/pdftex/`.

A.3.3 C++ and the Rcpp Package

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language. It was primarily oriented toward system programming, embedded resource-constrained software and large systems, providing a wide efficiency and flexibility of use. C++ is not very popular among statisticians if compared to R or Python programming languages but possess a wide number of libraries accounting for linear algebra and scientific computing tasks that are often required for implementing statistical software. Among those, in this PhD thesis we used the Armadillo C++ library (Sanderson and Curtin, 2016). Additional information can be found on the website: `https://isocpp.org`.

Sometimes, R code encounters bottlenecks that slow down the computational time. Performances can be sometimes improved by rewriting functions in C++ code, and executing them into a pure R environment: the connection is made by the fundamental Rcpp package (Eddelbuettel, 2013), which provides a user-friendly framework for addressing typical R bottlenecks including loops that can't be easily vectorized because subsequent iterations depend on previous ones, a typical problem when performing e.g. MCMC sampling or variational approximation optimizations. Additionally, the Armadillo library is accommodated within the Rcpp framework usage due to the RcppArmadillo package (Eddelbuettel and Sanderson, 2014). We refer to Eddelbuettel and Balamuta (2018) for an introductory tutorial.

A.3.4 Stan and the rstan Package

Stan (Carpenter *et al.*, 2017) is a state-of-the-art platform and probabilistic programming language for statistical modeling and high-performance statistical computations. It efficiently implements methods for full Bayesian statistical inference with MCMC (using the NUTS sampler of Hoffman and Gelman, 2014 and the Hamiltonian Monte Carlo sampler, see e.g. Neal, 2011), approximate Bayesian inference with variational approximations (employing the ADVI method of Kucukelbir *et al.*, 2017) and penalized maximum likelihood estimation with L-BFGS optimization. In all three cases, automatic differentiation is used to quickly and accurately evaluate gradients without burdening the user with the need to derive the partial derivatives. Additional information can be found on the official website: `https://mc-stan.org`.

Its wide popularity in Bayesian statistics is motivated by the fact that it allows sampling from the posterior distribution of a generic model, only requiring the

user to specify the model structure with a few simple lines of coding. The rstan library (Stan Development Team, 2020) provides an excellent interface for R with functions that parse, compile, test, estimate, and analyze Stan models by accessing the associated StanHeaders package.

A.3.5 TensorFlow and the tensorflow Package

TensorFlow (Abadi *et al.*, 2016) is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications. It was originally developed by researchers and engineers working on the Google Brain team within *Google's Machine Intelligence Research* organization to conduct machine learning and deep neural networks research. Additional information and resources can be found on the official website `https://www.tensorflow.org`.

TensorFlow provides stable Python and C++ APIs, as well as non-guaranteed backward compatible API for other languages. In R, the tensorflow package (Allaire and Tang, 2020) provides access to the complete TensorFlow API, see `https://tensorflow.rstudio.com`. Within this PhD thesis, we employ tensorflow to perform fast automatic differentiation via the GradientTape API: an introductory tutorial is given in `https://tensorflow.rstudio.com/tutorials/advanced/customization/autodiff/`.

Appendix B

This appendix contains additional details on definitions and derivations for implementing the algorithms described in Chapter 2. Some of them are adopted from Kim and Wand (2016), Kim and Wand (2018), Chen and Wand (2020) and Hall *et al.* (2020), to which we refer for further insights.

B.1 Function Definitions

B.1.1 Non-Analytic Integral Functions

Some integral-defined functions are required to express indefinite integrals as functions of some pre-specified parameters. From Section 2.1 of Kim and Wand (2016), we report

$$\mathcal{A}(p,q,r,s,t,u) \equiv \int_{-\infty}^{\infty} x^p \, \frac{\exp(qx - rx^2)}{(x^2 + sx + t)^u} \, \mathrm{d}x,$$

which is defined for $p \geq 0$, $q \in \mathbb{R}$, $r > 0$, $s \in \mathbb{R}$, $t > 1/4s^2$, $u > 0$ and

$$\mathcal{B}(p,q,r,s,t,u) \equiv \int_{-\infty}^{\infty} x^p \, \frac{\exp(qx - re^x - se^x/(t + e^x))}{(t + e^x)^u} \, \mathrm{d}x,$$

which is defined for $p \geq 0$, $q \in \mathbb{R}$, $r > 0$, $s \in \mathbb{R}$, $t > 0$, $u > 0$. Following Appendix 2.4 of Kim and Wand (2016), such integral-defined functions can be expressed in terms of this more convenient expressions:

$$\alpha\left(k, \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right) \equiv \mathcal{A}\left(k, a_1, -a_2, \frac{-2c_2}{c_1}, \frac{c3 - 2b_2}{c_1}, \frac{c_1 - 2b_1 - 2}{2}\right)$$

and

$$\beta\left(k,\ell,v,w,\left[\begin{array}{c}a_1\\a_2\end{array}\right],\left[\begin{array}{c}b_1\\b_2\end{array}\right],\left[\begin{array}{c}c_1\\c_2\\c_3\end{array}\right]\right)\equiv\mathcal{B}\left(k,\tfrac{\ell+c_1-1}{2}-a_1,\tfrac{c_1c_3-c_2^2}{2c_1}-a_2,-b_2\left(\tfrac{c_2}{c_1}+\tfrac{b_1}{2b_2}\right)^2,v,w\right).$$

Moreover, from Kim and Wand (2018) we report here

$$\mathcal{C}_b(p,q,r)\equiv\int_{-\infty}^{\infty}x^p\,\exp\Big(qx-rx^2-b(x)\Big)\,\mathrm{d}x$$

which is defined for $p\geq 0$, $q\in\mathbb{R}$, $r>0$ and $b:\mathbb{R}\longrightarrow\mathbb{R}$ is any function for which the resulting integral exists.

For the work presented in Chapter 2, we introduce:

$$\gamma_d\left(k,\left[\begin{array}{c}\boldsymbol{a}_1\\\boldsymbol{a}_2\end{array}\right],\left[\begin{array}{c}b_1\\b_2\end{array}\right],\left[\begin{array}{c}c_1\\c_2\\\boldsymbol{c}_3\\\boldsymbol{c}_4\end{array}\right]\right),\quad k\in\{0,1,2\}$$

defined, for $\boldsymbol{a}_1$ and $\boldsymbol{c}_3$ of dimension $d\times 1$ and $\boldsymbol{a}_2$ and $\boldsymbol{c}_4$ of dimension $d(d+1)/2\times 1$, as

$$\gamma_d\left(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\right)\equiv\int_{-\infty}^{\infty}\exp\left\{\left[\begin{array}{c}-x\\e^x\end{array}\right]^T\left(\left[\begin{array}{c}b_1\\b_2\end{array}\right]-\frac{1}{2}\left[\begin{array}{c}c_1\\c_2\end{array}\right]\right)+A_{\text{MVN}}\left(\left[\begin{array}{c}\boldsymbol{a}_1\\\boldsymbol{a}_2\end{array}\right]+e^x\left[\begin{array}{c}\boldsymbol{c}_3\\\boldsymbol{c}_4\end{array}\right]\right)-x\right\}\mathrm{d}x,$$

$$\begin{aligned}\gamma_d\left(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\right)\equiv&\int_{-\infty}^{\infty}\exp\left\{\left[\begin{array}{c}-x\\e^x\end{array}\right]^T\left(\left[\begin{array}{c}b_1\\b_2\end{array}\right]-\frac{1}{2}\left[\begin{array}{c}c_1\\c_2\end{array}\right]\right)+A_{\text{MVN}}\left(\left[\begin{array}{c}\boldsymbol{a}_1\\\boldsymbol{a}_2\end{array}\right]+e^x\left[\begin{array}{c}\boldsymbol{c}_3\\\boldsymbol{c}_4\end{array}\right]\right)-x\right\}\\&\times\left[-\frac{1}{2}\left\{\text{vech}^{-1}\left(\boldsymbol{a}_2+e^x\boldsymbol{c}_4\right)\right\}^{-1}\left\{\boldsymbol{a}_1+\frac{e^x}{2}\boldsymbol{c}_3\right\}\right]_1^d\mathrm{d}x\end{aligned}$$

and

$$\begin{aligned}\gamma_d\left(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\right)\equiv&\int_{-\infty}^{\infty}\exp\left\{\left[\begin{array}{c}-x\\e^x\end{array}\right]^T\left(\left[\begin{array}{c}b_1\\b_2\end{array}\right]-\frac{1}{2}\left[\begin{array}{c}c_1\\c_2\end{array}\right]\right)+A_{\text{MVN}}\left(\left[\begin{array}{c}\boldsymbol{a}_1\\\boldsymbol{a}_2\end{array}\right]+e^x\left[\begin{array}{c}\boldsymbol{c}_3\\\boldsymbol{c}_4\end{array}\right]\right)-x\right\}\\&\times\left[\frac{1}{4}\text{vech}\left\{\left[\text{vech}^{-1}\left(\boldsymbol{a}_2+e^x\boldsymbol{c}_4\right)\right]^{-1}\left\{\left[\boldsymbol{a}_1+\frac{e^x}{2}\boldsymbol{c}_3\right]\left[\boldsymbol{a}_1+\frac{e^x}{2}\boldsymbol{c}_3\right]^T\right.\right.\\&\left.\left.\times\left[\text{vech}^{-1}\left(\boldsymbol{a}_2+e^x\boldsymbol{c}_4\right)\right]^{-1}-2\boldsymbol{I}\right\}\right\}\Bigg]_1^{d(d+1)/2}\mathrm{d}x,\end{aligned}$$

where $A_{\text{MVN}}(\cdot)$ denotes the log-partition function of a multivariate d-dimensional Normal distribution and $[\cdot]_1^d$ is a shorthand notation emphasizing that the integral is actually a vector of d different integrals, each evaluated considering one of the d different elements of the vector contained in $[\cdot]$. Hence $\gamma_d(0, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is a scalar, $\gamma_d(1, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is a vector of length d and $\gamma_d(2, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is a vector of length $d(d+1)/2$, for suitably chosen vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$.

Moreover, we introduce:

$$\delta_d\left(k, \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \begin{bmatrix} c_1 \\ c_2 \\ \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{bmatrix}\right), \quad k \in \{0, 1, 2\}$$

defined, for $\boldsymbol{a}_1$ and $\boldsymbol{c}_3$ of dimension $d \times 1$ and $\boldsymbol{a}_2$ and $\boldsymbol{c}_4$ of dimension $d(d+1)/2 \times 1$, as

$$\delta_d(0, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \equiv \int_{-\infty}^{\infty} \exp\left\{ \begin{bmatrix} -x \\ e^x \end{bmatrix}^T \left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right) + A_{\text{MVN}}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix} + e^x \begin{bmatrix} \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{bmatrix} \right) - x \right\} \mathrm{d}x,$$

$$\delta_d(1, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \equiv \int_{-\infty}^{\infty} x \exp\left\{ \begin{bmatrix} -x \\ e^x \end{bmatrix}^T \left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right) + A_{\text{MVN}}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix} + e^x \begin{bmatrix} \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{bmatrix} \right) - x \right\} \mathrm{d}x,$$

and

$$\delta_d(2, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \equiv \int_{-\infty}^{\infty} \exp\left\{ \begin{bmatrix} -x \\ e^x \end{bmatrix}^T \left(\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \right) + A_{\text{MVN}}\left(\begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \end{bmatrix} + e^x \begin{bmatrix} \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{bmatrix} \right) \right\} \mathrm{d}x.$$

Practical `R` routines implementing numerical quadrature methods for computing $\mathcal{A}(p, q, r, s, t, u)$ and $\mathcal{B}(p, q, r, s, t, u)$ are given as supplementary material to Kim and Wand (2016) and are freely available in Matt P. Wand's personal webpage, see `http://matt-wand.utsacademics.info/KimWandCode.zip`. Although not used in our implementations, a recent work by Pogány and Nadarajah (2021) developed explicit forms for both integrals involving special mathematical functions, which can be computed with in-built routines available in classical programming languages.

Both $\mathcal{C}_b(p, q, r)$, $\gamma_d(k, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ and $\delta_d(k, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ only involves univariate integrals, and can be efficiently solved employing standard numerical quadrature methods. Practical advices for stable and efficient implementations are given Appendix B of Wand *et al.* (2011), bridging ideas of Laplace approximation and Gauss-Hermite quadrature for numerical resolution of complex integrals (Liu and Pierce, 1994).

B.1.2 Kullback-Leibler Projection Wrapper Functions

Kullback-Leibler projections are required in Chapter 2 for obtaining the natural parameter vectors $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a}$. As it will be shown later on in this Appendix, their computation usually involves univariate integrals to be solved numerically. In order to make the associated Power-EP update expressions appear in a more compact form, we illustrate hereafter some useful wrapper functions that will associate the non-analytic functions defined in the previous pages with optimal Kullback-Leibler projected natural parameter vectors.

From Appendix 2.4 of Kim and Wand (2016), we report below:

$$g(\ell, v, w, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \equiv (\log\text{–}\psi)^{-1}\left(\log\left\{\frac{\beta(0,\ell+1,v,w,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\beta(0,\ell-1,v,w,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right\} - \frac{\beta(1,\ell-1,v,w,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\beta(0,\ell-1,v,w,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right),$$

$$G^{\mathrm{N}}(\boldsymbol{a},\boldsymbol{b};\boldsymbol{c}) \equiv \left[\frac{\alpha(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\alpha(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})} - \left\{\frac{\alpha(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\alpha(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right\}^2\right]^{-1}\begin{bmatrix}\alpha(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})/\alpha(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\\ -1/2\end{bmatrix} - \boldsymbol{a},$$

$$G^{\mathrm{IG1}}\left(\boldsymbol{a},\begin{bmatrix}b_1\\ b_2\end{bmatrix};\begin{bmatrix}c_1\\ c_2\\ c_3\end{bmatrix}\right) \equiv \begin{bmatrix}-1-g(0,-2b_2/c_1,1/2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\\ \dfrac{-g(0,-2b_2/c_1,1/2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\beta(0,-1,-2b_2/c1,1/2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\beta(0,1,-2b_2/c_1,1/2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\end{bmatrix} - \boldsymbol{a}$$

and

$$G^{\mathrm{IG2}}\left(\boldsymbol{a},\begin{bmatrix}b_1\\ b_2\end{bmatrix};k\right) \equiv \begin{bmatrix}-1-g\left(k-2,-b_2,1-k/2-b_1,\boldsymbol{a},\begin{bmatrix}0\\ b_2\end{bmatrix},\begin{bmatrix}2\\0\\0\end{bmatrix}\right)\\ \dfrac{\left\{\begin{array}{l}-g\left(k-2,-b_2,1-k/2-b_1,\boldsymbol{a},\begin{bmatrix}0\\ b_2\end{bmatrix},\begin{bmatrix}2\\0\\0\end{bmatrix}\right)\\ \times\beta\left(0,k-3,-b_2,1-k/2-b_1,\boldsymbol{a},\begin{bmatrix}0\\ b_2\end{bmatrix},\begin{bmatrix}2\\0\\0\end{bmatrix}\right)\end{array}\right\}}{\beta\left(0,k-1,-b_2,1-k/2-b_1,\boldsymbol{a},\begin{bmatrix}0\\ b_2\end{bmatrix},\begin{bmatrix}2\\0\\0\end{bmatrix}\right)}\end{bmatrix} - \boldsymbol{a}.$$

For the work presented in Chapter 2, we introduce

$$G_{\text{lm}}^{\text{MVN}}(\boldsymbol{a},\boldsymbol{b};\boldsymbol{c}) \equiv \begin{bmatrix} \left\{ \dfrac{\text{vech}^{-1}\left(\gamma_d(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\right)}{\gamma_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})} - \left(\dfrac{\gamma_d(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\gamma_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right)\left(\dfrac{\gamma_d(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\gamma_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right)^T \right\}^{-1} \\ \times \left(\dfrac{\gamma_d(1,\boldsymbol{a},\boldsymbol{bc})}{\gamma_d(0,\boldsymbol{a},\boldsymbol{bc})}\right) \\ -\dfrac{1}{2}\text{vech}\left(\left\{ \dfrac{\text{vech}^{-1}\left(\gamma_d(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})\right)}{\gamma_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})} - \left(\dfrac{\gamma_d(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\gamma_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right)\left(\dfrac{\gamma_d(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\gamma_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right)^T \right\}^{-1} \right) \end{bmatrix} - \boldsymbol{a}$$

and

$$G^{\text{IG4}}(\boldsymbol{a},\boldsymbol{b};\boldsymbol{c}) \equiv \begin{bmatrix} -1 - (\log\text{–}\psi)^{-1}\left(\log\left\{\dfrac{\delta_d(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\delta_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right\} - \dfrac{\delta_d(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\delta_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right) \\ -(\log\text{–}\psi)^{-1}\left(\log\left\{\dfrac{\delta_d(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\delta_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right\} - \dfrac{\delta_d(1,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\delta_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}\right)\dfrac{\delta_d(0,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})}{\delta_d(2,\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})} \end{bmatrix} - \boldsymbol{a}.$$

Moreover, we generalize the $K_{\text{probit}}(\boldsymbol{a};c_0,\boldsymbol{c}_1)$ wrapper introduced in Definition 1 of Hall *et al.* (2020) into:

$$G_{\text{glm}}^{\text{MVN}}\left(\begin{bmatrix}\boldsymbol{a}_1\\ \boldsymbol{a}_2\end{bmatrix};c_0,\boldsymbol{c}_1,c_2,b\right) \equiv \begin{bmatrix}\boldsymbol{R}_5^T(\boldsymbol{a}_1 + r_3\boldsymbol{c}_1)\\ \text{vech}(\boldsymbol{R}_5^T\boldsymbol{A}_2)\end{bmatrix} - \boldsymbol{a},$$

where

$$\boldsymbol{A}_2 \equiv \text{vech}^{-1}(\boldsymbol{a}_2), \quad r_1 \equiv \boldsymbol{c}_1^T\boldsymbol{A}_2^{-1}\boldsymbol{c}_1, \quad r_2 \equiv (2c_0 - \boldsymbol{c}_1^T\boldsymbol{A}_2^{-1}\boldsymbol{a}_1)/r_1,$$

$$r_6 \equiv c_2 - r_2, \quad r_7 \equiv -r_1^{-1}, \quad r_3 \equiv r_2 + 2r_7\frac{\mathcal{C}_b(1,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)},$$

$$r_4 \equiv -2r_7^2\left(\frac{\mathcal{C}_b(2,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)} - \left(\frac{\mathcal{C}_b(1,r_6,r_7)}{\mathcal{C}_b(0,r_6,r_7)}\right)^2 + \frac{r_1}{2}\right), \quad \boldsymbol{R}_5 \equiv (\boldsymbol{A}_2 + r_4\boldsymbol{c}_1\boldsymbol{c}_1^T)^{-1}\boldsymbol{A}_2.$$

B.2 Derivations

B.2.1 Updates for the Univariate Normal Random Sample Model

We give explicit derivations of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a}$, together with compact representation of the associated $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}$ updates required for implementation of Algorithm 2.2. Derivations follow Appendix A.5 of Kim and Wand

(2016) with minor modifications to admit Power-EP approximation extension.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$

Following algebraic steps presented in Appendix A.5.2 of Kim and Wand (2016), the function of μ to be Kullback-Leibler-projected into the univariate Normal exponential family is

$$\begin{aligned} h(\mu) &\propto \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu}(\mu)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\mu\to\mathfrak{p}(\mu)}(\mu)\, \mathfrak{p}(\mu)^{\alpha} \\ &\propto \exp\left\{ \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix}^T \left((1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu} + \boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\mu)} + \alpha \begin{bmatrix} \mu_\mu/\sigma^2_\mu \\ -1/(2\sigma^2_\mu) \end{bmatrix} \right) \right\} \\ &\propto \exp\left\{ \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\leftrightarrow\mu} + \alpha \begin{bmatrix} \mu_\mu/\sigma^2_\mu \\ -1/(2\sigma^2_\mu) \end{bmatrix} \right) \right\}. \end{aligned}$$

Since it is proportional to a univariate Normal distribution, the Kullback-Leibler projection returns the exact natural parameters of $h(\mu)$. Hence

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\mu)\to\mu} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\leftrightarrow\mu} + \alpha \begin{bmatrix} \mu_\mu/\sigma^2_\mu \\ -1/(2\sigma^2_\mu) \end{bmatrix}$$

and

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\leftrightarrow\mu} + \alpha \begin{bmatrix} \mu_\mu/\sigma^2_\mu \\ -1/(2\sigma^2_\mu) \end{bmatrix} - \boldsymbol{\eta}^{(\alpha)}_{\mu\to\mathfrak{p}(\mu)} \longleftarrow \begin{bmatrix} \mu_\mu/\sigma^2_\mu \\ -1/(2\sigma^2_\mu) \end{bmatrix}.$$

Then $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\mu)\to\mu}$ is fixed at each iteration and corresponds to the natural parameter vector $\boldsymbol{\eta}_{\mathfrak{p}(\mu)}$ of $\mathfrak{p}(\mu)$.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$

Following algebraic steps presented in Appendix A.5.3 of Kim and Wand (2016), the function of μ to be Kullback-Leibler-projected into the univariate Normal exponential family is

$$\begin{aligned} h_{\blacklozenge}(\mu) \propto{}& \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}(\mu)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\mu\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}(\mu) \\ &\times \int_0^\infty \mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)^{\alpha} \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}(\sigma^2)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}(\sigma^2)\, \mathrm{d}\sigma^2. \end{aligned}$$

The integral can be solved analytically as

$$
\begin{aligned}
&\int_0^\infty \mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)^\alpha \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}(\sigma^2)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}(\sigma^2)\,\mathrm{d}\sigma^2 \\
&\propto \int_0^\infty \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\alpha \begin{bmatrix} -n/2 \\ -\|\boldsymbol{y}-\mu\mathbf{1}\|^2/2 \end{bmatrix} + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} + \boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)} \right) \right\} \mathrm{d}\sigma^2 \\
&\propto \int_0^\infty \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\alpha \begin{bmatrix} -n/2 \\ -\|\boldsymbol{y}-\mu\mathbf{1}\|^2/2 \end{bmatrix} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2} \right) \right\} \mathrm{d}\sigma^2 \\
&\propto \exp\left\{ A_{\mathrm{IG}}\left(\alpha \begin{bmatrix} -n/2 \\ -\|\boldsymbol{y}-\mu\mathbf{1}\|^2/2 \end{bmatrix} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2} \right) \right\} \\
&\propto \left(\alpha\frac{\|\boldsymbol{y}-\mu\mathbf{1}\|^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2 \right)^{-\alpha\frac{n}{2}+\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1+1},
\end{aligned}
$$

where $A_{\mathrm{IG}}(\cdot)$ denotes the log-partition function of an Inverse Gamma distribution (see Appendix A). Hence

$$
h_\blacklozenge(\mu) \propto \exp\left\{ \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu} \right\} \left(\alpha\frac{\|\boldsymbol{y}-\mu\mathbf{1}\|^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2 \right)^{-\alpha\frac{n}{2}+\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1+1}
$$

and doing some little algebra it is possible to express

$$
\begin{aligned}
\mu^*_\blacklozenge \equiv E_{h_\blacklozenge(\mu)}(\mu) &= \frac{\int_{-\infty}^{\infty} \mu\, h_\blacklozenge(\mu)\,\mathrm{d}\mu}{\int_{-\infty}^{\infty} h_\blacklozenge(\mu)\,\mathrm{d}\mu} \\
&= \frac{\mathcal{A}\left(1, \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2, -2\bar{y}, \frac{\sum_{i=1}^n y_i^2 - \frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2}{\alpha}}{n}, \frac{\alpha n - 2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1 - 2}{2}\right)}{\mathcal{A}\left(0, \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2, -2\bar{y}, \frac{\sum_{i=1}^n y_i^2 - \frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2}{\alpha}}{n}, \frac{\alpha n - 2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1 - 2}{2}\right)}
\end{aligned}
$$

and

$$
\begin{aligned}
(\mu^2)^*_\blacklozenge \equiv E_{h_\blacklozenge(\mu)}(\mu^2) &= \frac{\int_{-\infty}^{\infty} \mu^2\, h_\blacklozenge(\mu)\,\mathrm{d}\mu}{\int_{-\infty}^{\infty} h_\blacklozenge(\mu)\,\mathrm{d}\mu} \\
&= \frac{\mathcal{A}\left(2, \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2, -2\bar{y}, \frac{\sum_{i=1}^n y_i^2 - \frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2}{\alpha}}{n}, \frac{\alpha n - 2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1 - 2}{2}\right)}{\mathcal{A}\left(0, \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2, -2\bar{y}, \frac{\sum_{i=1}^n y_i^2 - \frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2}{\alpha}}{n}, \frac{\alpha n - 2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1 - 2}{2}\right)}.
\end{aligned}
$$

Straightforward application of Result 2.1 leads to

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu} \longleftarrow \begin{bmatrix} \mu^*_{\blacklozenge} \,/\, \left((\mu^2)^*_{\blacklozenge} - (\mu^*_{\blacklozenge})^2\right) \\ -1/\left(2(\mu^2)^*_{\blacklozenge} - (\mu^*_{\blacklozenge})^2\right) \end{bmatrix}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$ can be expressed in terms of the $G^{\mathrm{N}}(\boldsymbol{a},\boldsymbol{b};\boldsymbol{c})$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu} \longleftarrow G^{\mathrm{N}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}; \alpha \begin{bmatrix} n \\ \boldsymbol{y}^T\mathbf{1} \\ \|\boldsymbol{y}\|^2 \end{bmatrix}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$

Following algebraic steps presented in Appendix A.5.5 of Kim and Wand (2016), the function of σ^2 to be Kullback-Leibler-projected into the Inverse Gamma exponential family is

$$\begin{aligned} h_{\clubsuit}(\sigma^2) \propto & \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}(\sigma^2)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}(\sigma^2) \\ & \times \int_{-\infty}^{\infty} \mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)^{\alpha} \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\mu}(\mu)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\mu\to\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)}(\mu)\,\mathrm{d}\mu. \end{aligned}$$

With similar steps illustrated before, we get

$$\begin{aligned} h_{\clubsuit}(\sigma^2) \propto & \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2} \right\} \\ & \times (\sigma^2)^{(1-\alpha n)/2} \exp\left\{ -\frac{\alpha(n-1)s^2}{2\sigma^2} - \frac{\left(\bar{y} + \frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)^2}{2\left(\frac{\sigma^2}{\alpha n} - \frac{1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)} \right\} \left(\frac{\sigma^2}{\alpha n} - \frac{1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)^{-1/2}, \end{aligned}$$

and doing some little algebra it is possible to express

$$\log(\sigma^2)^*_{\clubsuit} \equiv E_{h_{\clubsuit}(\sigma^2)}(\log\sigma^2) = \frac{\int_0^{\infty} \log\sigma^2\, h_{\clubsuit}(\sigma^2)\,\mathrm{d}\sigma^2}{\int_0^{\infty} h_{\clubsuit}(\sigma^2)\,\mathrm{d}\sigma^2}$$

$$= -\frac{\mathcal{B}\left(1, \frac{\alpha n-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1, \frac{\alpha(n-1)s^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2 \left(\bar{y} + \frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)^2, -\frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}{\alpha n}, \frac{1}{2}\right)}{\mathcal{B}\left(0, \frac{\alpha n-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1, \frac{\alpha(n-1)s^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2 \left(\bar{y} + \frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)^2, -\frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}{\alpha n}, \frac{1}{2}\right)}$$

and

$$(1/\sigma^2)^*_{\clubsuit} \equiv E_{h_{\clubsuit}(\sigma^2)}(1/\sigma^2) = \frac{\int_0^\infty 1/\sigma^2\, h_{\clubsuit}(\sigma^2)\, \mathrm{d}\sigma^2}{\int_0^\infty h_{\clubsuit}(\sigma^2)\, \mathrm{d}\sigma^2}$$

$$= \frac{\mathcal{B}\left(0, \frac{\alpha n}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1, \frac{\alpha(n-1)s^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2 \left(\bar{y} + \frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)^2, -\frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}{\alpha n}, \frac{1}{2}\right)}{\mathcal{B}\left(0, \frac{\alpha n-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_1, \frac{\alpha(n-1)s^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2 \left(\bar{y} + \frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_1}{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}\right)^2, -\frac{2\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}\right]_2}{\alpha n}, \frac{1}{2}\right)}.$$

Straightforward application of Result 2.2 leads to

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} \longleftarrow \begin{bmatrix} -(\log\text{–}\psi)^{-1}\left\{\log((1/\sigma^2)^*_{\clubsuit}) + \log(\sigma^2)^*_{\clubsuit}\right\} - 1 \\ -(\log\text{–}\psi)^{-1}\left\{\log((1/\sigma^2)^*_{\clubsuit}) + \log(\sigma^2)^*_{\clubsuit}\right\} / (1/\sigma^2)^*_{\clubsuit} \end{bmatrix}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$ can be expressed in terms of the $G^{\text{IG1}}(\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{c})$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2} \longleftarrow G^{\text{IG1}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\sigma^2}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\leftrightarrow\mu}; \alpha \begin{bmatrix} n \\ \boldsymbol{y}^T\mathbf{1} \\ \|\boldsymbol{y}\|^2 \end{bmatrix}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\mu,\sigma^2)\to\sigma^2}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$

Following algebraic steps presented in Appendix A.5.4 of Kim and Wand (2016), the function of σ^2 to be Kullback-Leibler-projected into the Inverse Gamma exponential family is

$$h_{\spadesuit}(\sigma^2) \propto \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}(\sigma^2)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\sigma^2|a)}(\sigma^2) \times \int_0^\infty \mathfrak{p}(\sigma^2|a)^\alpha \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}(a)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{a\to\mathfrak{p}(\sigma^2|a)}(a)\, \mathrm{d}a.$$

With similar steps illustrated before, we get

$$h_{\spadesuit}(\sigma^2) \propto \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2} \right\} (\sigma^2)^{-3\alpha/2} \left(\frac{\alpha}{\sigma^2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2 \right)^{-\frac{\alpha}{2}+\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1+1},$$

and doing some little algebra it is possible to express

$$\begin{aligned}\log(\sigma^2)^*_{\spadesuit} \equiv E_{h_{\spadesuit}(\sigma^2)}(\log\sigma^2) &= \frac{\int_0^\infty \log\sigma^2\, h_{\spadesuit}(\sigma^2)\,\mathrm{d}\sigma^2}{\int_0^\infty h_{\spadesuit}(\sigma^2)\,\mathrm{d}\sigma^2} \\ &= -\frac{\mathcal{B}\left(1, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2}{\alpha}, -\frac{2-\alpha}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1\right)}{\mathcal{B}\left(0, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2}{\alpha}, -\frac{2-\alpha}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1\right)}\end{aligned}$$

and

$$\begin{aligned}(1/\sigma^2)^*_{\spadesuit} \equiv E_{h_{\spadesuit}(\sigma^2)}(1/\sigma^2) &= \frac{\int_0^\infty 1/\sigma^2\, h_{\spadesuit}(\sigma^2)\,\mathrm{d}\sigma^2}{\int_0^\infty h_{\spadesuit}(\sigma^2)\,\mathrm{d}\sigma^2} \\ &= \frac{\mathcal{B}\left(0, \frac{3\alpha}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2}{\alpha}, -\frac{2-\alpha}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1\right)}{\mathcal{B}\left(0, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y|\mu,\sigma^2)\leftrightarrow\sigma^2}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2}{\alpha}, -\frac{2-\alpha}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1\right)}.\end{aligned}$$

Straightforward application of Result 2.2 leads to

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \longleftarrow \begin{bmatrix} -(\log\text{–}\psi)^{-1}\left\{\log((1/\sigma^2)^*_{\spadesuit}) + \log(\sigma^2)^*_{\spadesuit}\right\} - 1 \\ -(\log\text{–}\psi)^{-1}\left\{\log((1/\sigma^2)^*_{\spadesuit}) + \log(\sigma^2)^*_{\spadesuit}\right\} / (1/\sigma^2)^*_{\spadesuit} \end{bmatrix}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$ can be expressed in terms of the $G^{\text{IG2}}(\boldsymbol{a},\boldsymbol{b};k)$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2} \longleftarrow G^{\text{IG2}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}, \begin{bmatrix} \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1 + 2(1-\alpha) \\ \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2 / \alpha \end{bmatrix}; 3\alpha\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$

Following algebraic steps presented in Appendix A.5.6 of Kim and Wand (2016), the function of σ^2 to be Kullback-Leibler-projected into the Inverse Gamma exponential family is

$$h_{\blacktriangledown}(a) \propto \exp\left\{ \begin{bmatrix} \log a \\ 1/a \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a} \right\} a^{-\alpha/2} \left(\frac{\alpha}{a} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_2 \right)^{-\frac{3\alpha}{2}+\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1+1},$$

and doing some little algebra it is possible to express

$$\begin{aligned}\log(a)^*_{\blacktriangledown} \equiv E_{h_{\blacktriangledown}(a)}(\log a) &= \frac{\int_0^\infty \log a\, h_{\blacktriangledown}(a)\,\mathrm{d}a}{\int_0^\infty h_{\blacktriangledown}(a)\,\mathrm{d}a} \\ &= -\frac{\mathcal{B}\left(1, \frac{\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_2}{\alpha}, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1\right)}{\mathcal{B}\left(0, \frac{\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_2}{\alpha}, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1\right)}\end{aligned}$$

and

$$\begin{aligned}(1/a)^*_{\blacktriangledown} \equiv E_{h_{\blacktriangledown}(a)}(1/a) &= \frac{\int_0^\infty 1/a\, h_{\blacktriangledown}(a)\,\mathrm{d}a}{\int_0^\infty h_{\blacktriangledown}(a)\,\mathrm{d}a} \\ &= \frac{\mathcal{B}\left(0, \frac{\alpha}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_2}{\alpha}, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1\right)}{\mathcal{B}\left(0, \frac{\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_1, -\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a}\right]_2, 0, -\frac{\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_2}{\alpha}, \frac{3\alpha-2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1\right)}.\end{aligned}$$

Straightforward application of Result 2.2 leads to

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \longleftarrow \begin{bmatrix} -(\log\text{–}\psi)^{-1}\left\{\log((1/a)^*_{\blacktriangledown}) + \log(a)^*_{\blacktriangledown}\right\} - 1 \\ -(\log\text{–}\psi)^{-1}\left\{\log((1/a)^*_{\blacktriangledown}) + \log(a)^*_{\blacktriangledown}\right\} / (1/a)^*_{\blacktriangledown} \end{bmatrix}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ can be expressed in terms of the $G^{\text{IG2}}(\boldsymbol{a}, \boldsymbol{b}; k)$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a} \longleftarrow G^{\text{IG2}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow a'}, \begin{bmatrix} \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_1 + 2(1-\alpha) \\ \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\leftrightarrow\sigma^2}\right]_2 / \alpha \end{bmatrix}; \alpha\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a'},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}$

Following algebraic steps presented in Appendix A.5.7 of Kim and Wand (2016), the function of a to be Kullback-Leibler-projected into the Inverse Gamma exponential family is

$$\begin{aligned} h(a) &\propto \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(a)\to a}(a)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{a\to\mathfrak{p}(a)}(a)\, \mathfrak{p}(\mu)^{\alpha} \\ &\propto \exp\left\{ \begin{bmatrix} \log a \\ 1/a \end{bmatrix}^T \left((1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} + \boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(a)} + \alpha \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix} \right) \right\} \\ &\propto \exp\left\{ \begin{bmatrix} \log a \\ 1/a \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\leftrightarrow a} + \alpha \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix} \right) \right\}. \end{aligned}$$

Since it is proportional to an Inverse Gamma distribution, the Kullback-Leibler projection returns the exact natural parameters of $h(a)$. Hence

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\leftrightarrow a} + \alpha \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix}$$

and

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\leftrightarrow a} + \alpha \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix} - \boldsymbol{\eta}^{(\alpha)}_{a\to\mathfrak{p}(a)} \longleftarrow \begin{bmatrix} -3/2 \\ -1/s^2 \end{bmatrix}.$$

Then $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}$ is fixed at each iteration, and corresponds to the natural parameter vector $\boldsymbol{\eta}_{\mathfrak{p}(a)}$ of $\mathfrak{p}(a)$.

B.2.2 Updates for the Normal Linear Regression Model

We give explicit derivations of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$, together with compact representation of the associated $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$ updates required for implementation of Algorithm 2.3. Derivations of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(a)\to a}$ and compact representation of the associated $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to\sigma^2}$, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\sigma^2|a)\to a}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(a)\to a}$ updates are identical to those given for the univariate Normal random sample model, to which we refer.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$

The function of $\boldsymbol{\beta}$ to be Kullback-Leibler-projected into the multivariate Normal exponential family is

$$\begin{aligned} h(\boldsymbol{\beta}) &\propto \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}(\boldsymbol{\beta})\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{\beta})}(\boldsymbol{\beta})\, \mathfrak{p}(\boldsymbol{\beta})^{\alpha} \\ &\propto \exp\left\{ \begin{bmatrix} \boldsymbol{\beta} \\ \text{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T) \end{bmatrix}^T \left((1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} + \boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{\beta})} + \alpha \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix} \right) \right\} \\ &\propto \exp\left\{ \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \alpha \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix} \right) \right\}. \end{aligned}$$

Since it is proportional to a multivariate Normal distribution, the Kullback-Leibler projection returns the exact natural parameters of $h(\boldsymbol{\beta})$. Hence

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \alpha \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix}$$

and

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \alpha \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix} - \boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{\beta})} \longleftarrow \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ -\frac{1}{2}\text{vech}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}) \end{bmatrix}.$$

Then $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ is fixed at each iteration, and corresponds to the natural parameter vector $\boldsymbol{\eta}_{\mathfrak{p}(\boldsymbol{\beta})}$ of $\mathfrak{p}(\boldsymbol{\beta})$.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$

The function of $\boldsymbol{\beta}$ to be Kullback-Leibler-projected into the multivariate Normal exponential family is

$$\begin{aligned} h_{\blacktriangledown}(\boldsymbol{\beta}) \propto{}& \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}(\boldsymbol{\beta})\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}(\boldsymbol{\beta}) \\ &\times \int_0^{\infty} \mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)^{\alpha} \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}(\sigma^2)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}(\sigma^2)\, \mathrm{d}\sigma^2. \end{aligned}$$

The integral can be solved analytically as

$$\int_0^{\infty} \mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)^{\alpha} \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}(\sigma^2)\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}(\sigma^2)\, \mathrm{d}\sigma^2$$

$$
\begin{aligned}
&\propto \int_0^\infty \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\alpha \begin{bmatrix} -n/2 \\ -\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2/2 \end{bmatrix} + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} + \boldsymbol{\eta}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)} \right) \right\} \mathrm{d}\sigma^2 \\
&\propto \int_0^\infty \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\alpha \begin{bmatrix} -n/2 \\ -\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2/2 \end{bmatrix} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} \right) \right\} \mathrm{d}\sigma^2 \\
&\propto \exp\left\{ A_{\mathrm{IG}} \left(\alpha \begin{bmatrix} -n/2 \\ -\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2/2 \end{bmatrix} + \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} \right) \right\} \\
&\propto \left(\alpha \frac{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} \right]_2 \right)^{-\alpha\frac{n}{2} + \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} \right]_1 + 1},
\end{aligned}
$$

and therefore

$$
h_{\blacktriangledown}(\boldsymbol{\beta}) \propto \exp\left\{ \begin{bmatrix} \boldsymbol{\beta} \\ \mathrm{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T) \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right\} \left(\alpha \frac{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|^2}{2} - \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} \right]_2 \right)^{-\alpha\frac{n}{2} + \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} \right]_1 + 1}.
$$

Kullback-Leibler projection onto the multivariate Normal exponential family following Result 2.3 would require the following three d-variate integrals to be computed, being a computationally intractable task as already discussed in Section 2.4:

$$
\int_{\mathbb{R}^d} h_{\blacktriangledown}(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta}, \qquad \int_{\mathbb{R}^d} \boldsymbol{\beta}\, h_{\blacktriangledown}(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta} \qquad \text{and} \qquad \int_{\mathbb{R}^d} \mathrm{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T)\, h_{\blacktriangledown}(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta}.
$$

Nevertheless, it is still possible to proceed inverting the integration order: first we integrate over $\mathrm{d}\boldsymbol{\beta}$, and then over $\mathrm{d}\sigma^2$. This yields univariate numerical integrals to be solved, since all the d-divariate $\mathrm{d}\boldsymbol{\beta}$ integrals admit explicit solutions. Let

$$
\boldsymbol{\beta}^{\otimes k} \equiv \begin{cases} 1 & \text{if } k = 0 \\ \boldsymbol{\beta} & \text{if } k = 1 \\ \mathrm{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T) & \text{if } k = 2 \end{cases},
$$

then application of Result 2.3 requires the following quantities to be computed:

$$
\mathcal{M}_k^{\blacktriangledown} = \int_{\mathbb{R}^d} \boldsymbol{\beta}^{\otimes k}\, h_{\blacktriangledown}(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta} \quad \text{for} \quad k \in \{0,1,2\}.
$$

With small algebraic steps, it is possible to show that

$$
\mathcal{M}_k^{\blacktriangledown} \propto \int_0^\infty \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} - \frac{\alpha}{2} \begin{bmatrix} n \\ \|\boldsymbol{y}\|^2 \end{bmatrix} \right) \right\}
$$

$$\times \int_{\mathbb{R}^d} \boldsymbol{\beta}^{\otimes k} \exp\left\{ \begin{bmatrix} \boldsymbol{\beta} \\ \text{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T) \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} + \frac{\alpha}{\sigma^2} \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix} \right) \right\} \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\sigma^2.$$

Now recall results from exponential family distributions recalled in Appendix A.2, yielding explicit expressions for the expected value of the natural sufficient statistic vector of a multivariate Normal distribution and for its normalizing constant. It follows that

$$\begin{aligned} \mathcal{M}_0^{\blacktriangledown} &\propto \int_0^{\infty} \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} - \frac{\alpha}{2} \begin{bmatrix} n \\ \|\boldsymbol{y}\|^2 \end{bmatrix} \right) \right. \\ &\qquad \left. + A_{\text{MVN}} \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} + \frac{\alpha}{\sigma^2} \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix} \right) \right\} \mathrm{d}\sigma^2 \\ &\propto \gamma_d \left(0, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}'}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \alpha\boldsymbol{D} \right), \end{aligned}$$

$$\begin{aligned} \mathcal{M}_1^{\blacktriangledown} &\propto \int_0^{\infty} \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} - \frac{\alpha}{2} \begin{bmatrix} n \\ \|\boldsymbol{y}\|^2 \end{bmatrix} \right) + A_{\text{MVN}} \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} + \frac{\alpha}{\sigma^2} \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix} \right) \right\} \\ &\qquad \times \left[-\frac{1}{2} \left\{ \text{vech}^{-1}\left(\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right]_2 - \frac{\alpha}{2\sigma^2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \right) \right\}^{-1} \left\{ \left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right]_1 + \frac{\alpha}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y} \right\} \right]_1^d \mathrm{d}\sigma^2 \\ &\propto \gamma_d \left(1, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}'}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \alpha\boldsymbol{D} \right) \end{aligned}$$

and

$$\begin{aligned} \mathcal{M}_2^{\blacktriangledown} &\propto \int_0^{\infty} \exp\left\{ \begin{bmatrix} \log\sigma^2 \\ 1/\sigma^2 \end{bmatrix}^T \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2} - \frac{\alpha}{2} \begin{bmatrix} n \\ \|\boldsymbol{y}\|^2 \end{bmatrix} \right) + A_{\text{MVN}} \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} + \frac{\alpha}{\sigma^2} \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix} \right) \right\} \\ &\qquad \times \left[\frac{1}{4}\text{vech}\left\{ \left[\text{vech}^{-1}\left(\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right]_2 - \frac{\alpha}{2\sigma^2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \right) \right]^{-1} \left\{ \left[\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right]_1 + \frac{\alpha}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y} \right] \right. \right. \\ &\qquad \left. \left. \times \left[\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right]_1 + \frac{\alpha}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y} \right]^T \left[\text{vech}^{-1}\left(\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}} \right]_2 - \frac{\alpha}{2\sigma^2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \right) \right]^{-1} - 2\boldsymbol{I} \right\} \right]_1^{\frac{d(d+1)}{2}} \mathrm{d}\sigma^2 \\ &\propto \gamma_d \left(2, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}'}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \alpha\boldsymbol{D} \right), \end{aligned}$$

where $\boldsymbol{D} = (n, \|\boldsymbol{y}\|^2, (\boldsymbol{X}^T\boldsymbol{y})^T, -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X})^T)^T$. Now, let

$$\boldsymbol{\mu}^*_{\blacktriangledown} = \frac{\mathcal{M}_1^{\blacktriangledown}}{\mathcal{M}_0^{\blacktriangledown}} \quad \text{and} \quad \boldsymbol{\Sigma}^*_{\blacktriangledown} = \frac{\text{vech}^{-1}(\mathcal{M}_2^{\blacktriangledown})}{\mathcal{M}_0^{\blacktriangledown}} - \left(\frac{\mathcal{M}_1^{\blacktriangledown}}{\mathcal{M}_0^{\blacktriangledown}} \right) \left(\frac{\mathcal{M}_1^{\blacktriangledown}}{\mathcal{M}_0^{\blacktriangledown}} \right)^T.$$

Then, application of Result 2.3 leads to the natural parameter vector of the Kullback-Leibler projection of $h_{\blacktriangledown}(\boldsymbol{\beta})$ into the multivariate Normal exponential family of distributions being:

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} \longleftarrow \begin{bmatrix} (\boldsymbol{\Sigma}^*_{\blacktriangledown})^{-1}\boldsymbol{\mu}^*_{\blacktriangledown} \\ -\frac{1}{2}\text{vech}\left((\boldsymbol{\Sigma}^*_{\blacktriangledown})^{-1}\right) \end{bmatrix},$$

and from application of (2.10), we get

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} \longleftarrow \boldsymbol{\eta}^{*}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} - \boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ can be expressed in terms of the $G^{\text{MVN}}_{\text{lm}}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})$ wrapper as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}} \longleftarrow G^{\text{MVN}}_{\text{lm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2};\ \alpha\begin{bmatrix} n \\ \|\boldsymbol{y}\|^2 \\ \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}$ vector appearing on the right-hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$

The function of σ^2 to be Kullback-Leibler-projected into the Inverse Gamma exponential family is

$$\begin{aligned} h_{\blacktriangle}(\sigma^2) &\propto \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}(\sigma^2)\right)^{1-\alpha}\mathfrak{m}^{(\alpha)}_{\sigma^2\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}(\sigma^2) \\ &\quad\times\int_{-\infty}^{\infty}\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)^{\alpha}\left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\boldsymbol{\beta}}(\boldsymbol{\beta})\right)^{1-\alpha}\mathfrak{m}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)}(\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\beta} \\ &\propto \exp\left\{\begin{bmatrix}\log\sigma^2 \\ 1/\sigma^2\end{bmatrix}^T\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}\right\}(2\pi)^{-(n\alpha-d)^2}\exp\left\{A_{\text{MVN}}\left(\frac{\alpha}{\sigma^2}\begin{bmatrix}\boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\text{vech}(\boldsymbol{X}^T\boldsymbol{X})\end{bmatrix}\right)\right\}. \end{aligned}$$

Doing some little algebra it is possible to express

$$\log(\sigma^2)^*_{\blacktriangle} \equiv E_{h_{\blacktriangle}(\sigma^2)}(\log\sigma^2) = \frac{\int_0^{\infty}\log(\sigma^2)h_{\blacktriangle}(\sigma^2)\,\mathrm{d}\sigma^2}{\int_0^{\infty}h_{\blacktriangle}(\sigma^2)\,\mathrm{d}\sigma^2} = -\frac{\delta_d\left(1,\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2},\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}};\alpha\boldsymbol{D}\right)}{\delta_d\left(0,\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2},\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}};\alpha\boldsymbol{D}\right)}$$

and

$$(1/\sigma^2)^*_{\blacktriangle} \equiv E_{h_{\blacktriangle}(\sigma^2)}(1/\sigma^2) = \frac{\int_0^\infty (1/\sigma^2) h_{\blacktriangle}(\sigma^2)\,\mathrm{d}\sigma^2}{\int_0^\infty h_{\blacktriangle}(\sigma^2)\,\mathrm{d}\sigma^2} = \frac{\delta_d\left(2, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}}; \alpha\boldsymbol{D}\right)}{\delta_d\left(0, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}}; \alpha\boldsymbol{D}\right)}$$

where again $\boldsymbol{D} = (n, \|\boldsymbol{y}\|^2, (\boldsymbol{X}^T\boldsymbol{y})^T, -\frac{1}{2}\mathrm{vech}(\boldsymbol{X}^T\boldsymbol{X})^T)^T$. Straightforward application of Result 2.2 leads to

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} \longleftarrow \begin{bmatrix} -(\log\text{–}\psi)^{-1}\left\{\log((1/\sigma^2)^*_{\blacktriangle}) + \log(\sigma^2)^*_{\blacktriangle}\right\} - 1 \\ -(\log\text{–}\psi)^{-1}\left\{\log((1/\sigma^2)^*_{\blacktriangle}) + \log(\sigma^2)^*_{\blacktriangle}\right\} / (1/\sigma^2)^*_{\blacktriangle} \end{bmatrix}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$ can be expressed in terms of the $G^{\mathrm{IG4}}(\boldsymbol{a},\boldsymbol{b};\boldsymbol{c})$ wrapper as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2} \longleftarrow G^{\mathrm{IG4}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\sigma^2}, \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\leftrightarrow\boldsymbol{\beta}}; \alpha \begin{bmatrix} n \\ \|\boldsymbol{y}\|^2 \\ \boldsymbol{X}^T\boldsymbol{y} \\ -\frac{1}{2}\mathrm{vech}(\boldsymbol{X}^T\boldsymbol{X}) \end{bmatrix}\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2)\to\sigma^2}$ vector appearing on the right-hand side of the update expression refers to its value in the previous iteration.

B.2.3 Updates for Some Notable GLMs

We give explicit derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$, together with compact representation of the associated $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ update required for implementation of Algorithm 2.4, for each of the three different notable GLMs selected, for any $1 \leq i \leq n$. Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and compact representation of the associated $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ are identical to those given for the Normal linear regression model regardless of the selected GLM type, to which we refer.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ for the Probit Regression Model

The function of $\boldsymbol{\beta}$ to be Kullback-Leibler projected into the multivariate Normal exponential family distribution is:

$$\begin{aligned} h_{\bigstar}(\boldsymbol{\beta}) &\propto \left(\mathfrak{m}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}(\boldsymbol{\beta})\right)^{1-\alpha} \mathfrak{m}^{(\alpha)}_{\boldsymbol{\beta}\to\mathfrak{p}(y_i|\boldsymbol{\beta})}(\boldsymbol{\beta})\,\mathfrak{p}(y_i|\boldsymbol{\beta})^{\alpha} \\ &\propto \exp\left\{ \begin{bmatrix} \boldsymbol{\beta} \\ \mathrm{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T) \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right\} \Phi\left((2y_i - 1)\boldsymbol{x}_i^T\boldsymbol{\beta}\right)^{\alpha} \end{aligned}$$

$$= \exp\left\{ \begin{bmatrix} \boldsymbol{\beta} \\ \text{vech}(\boldsymbol{\beta}\boldsymbol{\beta}^T) \end{bmatrix}^T \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right\} \Phi\left(c_{0i} + \boldsymbol{c}_{1i}^T\boldsymbol{\beta}\right)^{\alpha},$$

where we define $c_{0i} = 0$ and $\boldsymbol{c}_{1i} = (2y_i - 1)\boldsymbol{x}_i$. Let

$$\boldsymbol{\beta}^{\otimes k} \equiv \begin{cases} 1 & \text{if } k = 0 \\ \boldsymbol{\beta} & \text{if } k = 1 \\ \boldsymbol{\beta}\boldsymbol{\beta}^T & \text{if } k = 2 \end{cases},$$

then Result 2.3 requires the following quantities to be computed

$$\mathcal{M}_k^{\bigstar} = \int_{\mathbb{R}^d} \boldsymbol{\beta}^{\otimes k}\, h_{\bigstar}(\boldsymbol{\beta})\, \mathrm{d}\boldsymbol{\beta} \qquad \text{for } k \in \{0, 1, 2\},$$

which are computationally intractable even for small values of d. Now, let define

$$\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \equiv -\frac{1}{2}\left\{\text{vech}^{-1}\left(\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right]_2\right)\right\}^{-1} \text{ and } \quad \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \equiv \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\left[\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right]_1$$

and operate the change of variable $\boldsymbol{z} = \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{-1/2}\left(\boldsymbol{\beta} - \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)$, from which $\mathcal{M}_k^{\bigstar}$ can be rewritten in more convenient form as:

$$\begin{aligned}\mathcal{M}_k^{\bigstar} = (2\pi)^{d/2} \exp\left\{A_{\text{MVN}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)\right\} \int_{\mathbb{R}^d} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2}\boldsymbol{z}\right)^{\otimes k} \\ \times \Phi\left(c_{0i} + \boldsymbol{c}_{1i}^T\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left\{\left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2}\boldsymbol{c}_{1i}\right\}^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}.\end{aligned}$$

After discarding the normalizing constant, which is the same for each $k = 0, 1, 2$, a general expression reads:

$$\begin{aligned}\mathcal{M}_k^{\bigstar} \propto \int_{\mathbb{R}^d} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2}\boldsymbol{z}\right)^{\otimes k} \\ \times \Phi\left(c_{0i} + \boldsymbol{c}_{1i}^T\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left\{\left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2}\boldsymbol{c}_{1i}\right\}^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}.\end{aligned}$$

The computation of this d-variate integrals can be boiled down into univariate integrals employing Result 2.3, with

$$a_i = c_{0i} + \boldsymbol{c}_{1i}^T\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \qquad \text{and} \qquad \boldsymbol{b}_i = \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2}\boldsymbol{c}_{1i}.$$

In particular,

$$\mathcal{M}_0^{\bigstar} \propto \int_{\mathbb{R}^d} \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} = \int_{-\infty}^{\infty} \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha} \phi(z)\, \mathrm{d}z,$$

$$\begin{aligned}
\mathcal{M}_1^{\bigstar} &\propto \int_{\mathbb{R}^d} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \boldsymbol{z}\right) \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\
&= \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \int_{\mathbb{R}^d} \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \int_{\mathbb{R}^d} \boldsymbol{z}\, \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\
&= \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \int_{-\infty}^{\infty} \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha} \phi(z)\, \mathrm{d}z \\
&\quad + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \frac{\boldsymbol{b}_i}{\|\boldsymbol{b}_i\|} \int_{-\infty}^{\infty} z\, \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha} \phi(z)\, \mathrm{d}z
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{M}_2^{\bigstar} &\propto \int_{\mathbb{R}^d} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \boldsymbol{z}\right) \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \boldsymbol{z}\right)^T \\
&\quad \times \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\
&= \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^T \int_{\mathbb{R}^d} \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\
&\quad + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \left(\int_{\mathbb{R}^d} \boldsymbol{z}\, \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}\right) \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^T \\
&\quad + \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \left(\int_{\mathbb{R}^d} \boldsymbol{z}\, \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}\right)^T \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \\
&\quad + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \left(\int_{\mathbb{R}^d} \boldsymbol{z}\boldsymbol{z}^T\, \Phi\left(a_i + \boldsymbol{b}_i^T \boldsymbol{z}\right)^{\alpha} \phi(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}\right) \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \\
&= \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^T \int_{-\infty}^{\infty} \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha} \phi(z)\, \mathrm{d}z + \\
&\quad + \frac{2}{\|\boldsymbol{b}_i\|} \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \boldsymbol{b}_i \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^T \int_{-\infty}^{\infty} z\, \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha} \phi(z)\, \mathrm{d}z \\
&\quad + \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \int_{-\infty}^{\infty} \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha}\, \phi(z)\, \mathrm{d}z + \\
&\quad\quad + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \frac{\boldsymbol{b}_i \boldsymbol{b}_i^T}{\|\boldsymbol{b}_i\|^2} \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\right)^{1/2} \int_{-\infty}^{\infty} (z^2 - 1) \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha}\, \phi(z)\, \mathrm{d}z.
\end{aligned}$$

Therefore, evaluation of $\mathcal{M}_0^{\bigstar}$, $\mathcal{M}_1^{\bigstar}$ and $\mathcal{M}_2^{\bigstar}$ then only requires the computation of three univariate integrals. Operating the change of variable $x = a_i + \|\boldsymbol{b}_i\| z$, they can be expressed as:

$$\int_{-\infty}^{\infty} \Phi(a_i + \|\boldsymbol{b}_i\| z)^{\alpha} \phi(z)\, \mathrm{d}z = \frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^2} \exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\} \mathcal{C}_{-\alpha \log \Phi(x)}\left(0, \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right),$$

$$\int_{-\infty}^{\infty} x\,\Phi(a_i+\|\boldsymbol{b}_i\|z)^{\alpha}\phi(z)\,\mathrm{d}z = \frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^3}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\}$$
$$\times\left(\mathcal{C}_{-\alpha\log\Phi(x)}\left(1,\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)-a_i\mathcal{C}_{-\alpha\log\Phi(x)}\left(0,\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right),$$

and

$$\int_{-\infty}^{\infty} x^2\,\Phi(a_i+\|\boldsymbol{b}_i\|z)^{\alpha}\phi(z)\,\mathrm{d}z = \frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^4}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\}$$
$$\times\left\{\mathcal{C}_{-\alpha\log\Phi(x)}\left(2,\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)-2a_i\mathcal{C}_{-\alpha\log\Phi(x)}\left(1,\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right.$$
$$\left.+a_i^2\mathcal{C}_{-\alpha\log\Phi(x)}\left(0,\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right\}.$$

Their evaluation is straightforward once the univariate numerical integrals

$$\mathcal{C}_{-\alpha\log\Phi(x)}\left(k,\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)$$

are evaluated for each $k=0,1,2$. Just recall the general definition of $\mathcal{C}_b(p,q,r)$.

Now, let

$$b_\alpha(x) = -\alpha\log\Phi(x),$$
$$r_{i1} = -2\|\boldsymbol{b}_i\|^2 = -2\boldsymbol{c}_{1i}^T\,\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\,\boldsymbol{c}_{1i},$$
$$r_{i2} = -a_i/\|\boldsymbol{b}_i\|^2 = 2(c_{0i}+\boldsymbol{c}_{1i}^T\,\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}})/r_{i1},$$
$$r_{i6} = a_i/\|\boldsymbol{b}_i\|^2 = -r_{i2}, \qquad r_{i7} = 1/(2\|\boldsymbol{b}_i\|^2) = -r_{i1}^{-1}$$

and descard the normalizing constant $(2\pi)^{-1/2}\exp(-a_i^2/(2\|\boldsymbol{b}_i\|^2))/\|\boldsymbol{b}_i\|^2$ which is the same for each $k=0,1,2$. Simple algebraic manipulations allow to express $\mathcal{M}_k^{\bigstar}$ in terms of $\mathcal{C}_{b_\alpha}(k,r_{i6},r_{i7})$ for each $k=0,1,2$, namely:

$$\mathcal{M}_0^{\bigstar} \propto \mathcal{C}_{b_\alpha}(0,r_{i6},r_{i7}),$$

$$\mathcal{M}_1^{\bigstar} \propto \mathcal{C}_{b_\alpha}(0,r_{i6},r_{i7})\,\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + (2r_{i7}\,\mathcal{C}_{b_\alpha}(1,r_{i6},r_{i7}) + r_{i2}\,\mathcal{C}_{b_\alpha}(0,r_{i6},r_{i7}))\,\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}\,\boldsymbol{c}_{1i},$$

and

$$\begin{aligned}\mathcal{M}_2^{\bigstar} \propto\ & \mathcal{C}_{b_\alpha}(0, r_{i6}, r_{i7}) \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right)^T + \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right) \\ & + \left(r_{i2}\, \mathcal{C}_{b_\alpha}(0, r_{i6}, r_{i7}) + 2 r_{i7}\, \mathcal{C}_{b_\alpha}(1, r_{i6}, r_{i7}) \right) \\ & \quad \times \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \boldsymbol{c}_{1i} \left(\boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right)^T + \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \boldsymbol{c}_{1i}^T \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right) \\ & + \left(4 r_{i7}^2\, \mathcal{C}_{b_\alpha}(2, r_{i6}, r_{i7}) + 4 r_{i7} r_{i2}\, \mathcal{C}_{b_\alpha}(1, r_{i6}, r_{i7}) + (r_{i2}^2 - r_{i7})\, \mathcal{C}_{b_\alpha}(0, r_{i6}, r_{i7}) \right) \\ & \quad + \left(\boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \boldsymbol{c}_{1i} \boldsymbol{c}_{i1}^T \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \right).\end{aligned}$$

Now, let

$$\boldsymbol{\mu}^*_{\bigstar} = \frac{\mathcal{M}_1^{\bigstar}}{\mathcal{M}_0^{\bigstar}} = \boldsymbol{\mu}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + \left(2 r_{i7} \frac{\mathcal{C}_{b_\alpha}(1, r_{i6}, r_{i7})}{\mathcal{C}_{b_\alpha}(0, r_{i6}, r_{i7})} + r_{i2} \right) \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \boldsymbol{c}_{1i}$$

and

$$\begin{aligned}\boldsymbol{\Sigma}^*_{\bigstar} &= \frac{\mathcal{M}_2^{\bigstar}}{\mathcal{M}_0^{\bigstar}} - \left(\frac{\mathcal{M}_1^{\bigstar}}{\mathcal{M}_0^{\bigstar}} \right) \left(\frac{\mathcal{M}_1^{\bigstar}}{\mathcal{M}_0^{\bigstar}} \right)^T \\ &= \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} + (2 r_{i7})^2 \left(\frac{\mathcal{C}_{b_\alpha}(2, r_{i6}, r_{i7})}{\mathcal{C}_{b_\alpha}(0, r_{i6}, r_{i7})} - \left\{ \frac{\mathcal{C}_{b_\alpha}(1, r_{i6}, r_{i7})}{\mathcal{C}_{b_\alpha}(0, r_{i6}, r_{i7})} \right\}^2 + \frac{r_{i1}}{2} \right) \\ &\quad \times \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}} \boldsymbol{c}_{1i} \boldsymbol{c}_{1i}^T \boldsymbol{\Sigma}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}.\end{aligned}$$

Then, application of Result 2.3 leads to the natural parameter vector of the Kullback-Leibler projection of $h_{\bigstar}(\boldsymbol{\beta})$ into the multivariate Normal exponential family of distributions being:

$$\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\rightarrow\boldsymbol{\beta}} \longleftarrow \begin{bmatrix} (\boldsymbol{\Sigma}^*_{\bigstar})^{-1} \boldsymbol{\mu}^*_{\bigstar} \\ -\frac{1}{2} \text{vech}\left((\boldsymbol{\Sigma}^*_{\bigstar})^{-1} \right) \end{bmatrix},$$

and from application of (2.10), we get

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\rightarrow\boldsymbol{\beta}} \longleftarrow \boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\rightarrow\boldsymbol{\beta}} - \boldsymbol{\eta}^{(\alpha)}_{\boldsymbol{\beta}\rightarrow\mathfrak{p}(y_i|\boldsymbol{\beta})}.$$

Finally, $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\rightarrow\boldsymbol{\beta}}$ can be expressed in terms of the $G^{\text{MVN}}_{\text{glm}}(\boldsymbol{a}; c_0, \boldsymbol{c}_1, c_2, b)$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\rightarrow\boldsymbol{\beta}} \longleftarrow G^{\text{MVN}}_{\text{glm}} \left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}; 0, (2y_i - 1)\boldsymbol{x}_i, 0, -\alpha \log(\Phi(x)) \right) + (1 - \alpha) \boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\rightarrow\boldsymbol{\beta}},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ for the Logistic Regression Model

Following similar steps showed before for the probit regression model, with $\Phi((2y_i-1)\boldsymbol{x}_i^T\boldsymbol{\beta})$ replaced by the logistic regression marginal likelihood $\text{expit}((2y_i-1)\boldsymbol{x}_i^T\boldsymbol{\beta})$, the following three integrals arise:

$$\int_{-\infty}^{\infty}\text{expit}(a_i+\|\boldsymbol{b}_i\|z)^{\alpha}\phi(z)\,\text{d}z=\frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^2}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\}\mathcal{C}_{\alpha\log(1+e^x)}\left(0,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right),$$

$$\begin{aligned}\int_{-\infty}^{\infty}x\,\text{expit}(a_i+\|\boldsymbol{b}_i\|z)^{\alpha}\phi(z)\,\text{d}z&=\frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^3}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\}\\&\times\left(\mathcal{C}_{\alpha\log(1+e^x)}\left(1,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)-a_i\mathcal{C}_{\alpha\log(1+e^x)}\left(0,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right),\end{aligned}$$

and

$$\begin{aligned}\int_{-\infty}^{\infty}x^2\,\text{expit}(a_i+\|\boldsymbol{b}_i\|z)^{\alpha}\phi(z)\,\text{d}z&=\frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^4}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\}\left(\mathcal{C}_{\alpha\log(1+e^x)}\left(2,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right.\\&\left.-2a_i\mathcal{C}_{\alpha\log(1+e^x)}\left(1,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)+a_i^2\mathcal{C}_{\alpha\log(1+e^x)}\left(0,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right).\end{aligned}$$

Their evaluation is straightforward one the univariate numerical integrals

$$\mathcal{C}_{\alpha\log(1+e^x)}\left(k,\alpha+\frac{a_i}{\|\boldsymbol{b}_i\|^2},\frac{1}{2\|\boldsymbol{b}_i\|^2}\right)$$

are evaluated for each $k=0,1,2$. After substituting into the expressions given for the derivation of the probit regression model natural parameter vector updates, with $b_\alpha(x)=\alpha\log(1+e^x)$ and $r_{i6}=\alpha-r_{i2}$, then $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ can be expressed in terms of the $G^{\text{MVN}}_{\text{glm}}(\boldsymbol{a};c_0,\boldsymbol{c}_1,c_2,b)$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}\longleftarrow G^{\text{MVN}}_{\text{glm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}};0,(2y_i-1)\boldsymbol{x}_i,\alpha,\alpha\log(1+e^x)\right)+(1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}'},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Derivation of $\boldsymbol{\eta}^{*(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ and $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(\boldsymbol{\beta})\to\boldsymbol{\beta}}$ for the Poisson Regression Model

Following similar steps showed before for the probit regression model, with $\Phi((2y_i-1)\boldsymbol{x}_i^T\boldsymbol{\beta})$ replaced by the Poisson regression marginal likelihood $\exp\{y_i\boldsymbol{x}_i^T\boldsymbol{\beta} - \exp(\boldsymbol{x}_i^T\boldsymbol{\beta}) - \log(y_i!)\}$, $\boldsymbol{c}_{1i} = \boldsymbol{x}_i$ and $c_{2i} = y_i$, the following three integrals arise:

$$\int_{-\infty}^{\infty} \exp\{c_{2i}(a_i + \|\boldsymbol{b}_i\|z) - \exp(a_i + \|\boldsymbol{b}_i\|z)\}^{\alpha}\phi(z)\,\mathrm{d}z = \frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^2}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\} \times \mathcal{C}_{\alpha e^x}\left(0, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right),$$

$$\int_{-\infty}^{\infty} x\,\exp\{c_{2i}(a_i + \|\boldsymbol{b}_i\|z) - \exp(a_i + \|\boldsymbol{b}_i\|z)\}^{\alpha}\phi(z)\,\mathrm{d}z = \frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^3}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\} \times \left(\mathcal{C}_{\alpha e^x}\left(1, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right) - a_i\mathcal{C}_{\alpha e^x}\left(0, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right),$$

and

$$\int_{-\infty}^{\infty} x^2\,\exp\{c_{2i}(a_i + \|\boldsymbol{b}_i\|z) - \exp(a_i + \|\boldsymbol{b}_i\|z)\}^{\alpha}\phi(z)\,\mathrm{d}z = \frac{(2\pi)^{-1/2}}{\|\boldsymbol{b}_i\|^4}\exp\left\{-\frac{a_i^2}{2\|\boldsymbol{b}_i\|^2}\right\} \times \left(\mathcal{C}_{\alpha e^x}\left(2, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right) - 2a_i\mathcal{C}_{\alpha e^x}\left(1, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right) + a_i^2\mathcal{C}_{\alpha e^x}\left(0, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right)\right).$$

Their evaluation is straightforward one the univariate numerical integrals

$$\mathcal{C}_{\alpha e^x}\left(k, \alpha c_{2i} + \frac{a_i}{\|\boldsymbol{b}_i\|^2}, \frac{1}{2\|\boldsymbol{b}_i\|^2}\right)$$

are evaluated for each $k = 0, 1, 2$. After substituting into the expressions given for the derivation of the probit regression model natural parameter vector updates, with $b_\alpha(x) = \alpha e^x$ and $r_{i6} = \alpha c_{2i} - r_{i2}$, then $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ can be expressed in terms of the $G^{\text{MVN}}_{\text{glm}}(\boldsymbol{a}; c_0, \boldsymbol{c}_1, c_2, b)$ wrapper function as

$$\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}} \longleftarrow G^{\text{MVN}}_{\text{glm}}\left(\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\leftrightarrow\boldsymbol{\beta}}; 0, \boldsymbol{x}_i, \alpha y_i, \alpha e^x\right) + (1-\alpha)\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}},$$

where the $\boldsymbol{\eta}^{(\alpha)}_{\mathfrak{p}(y_i|\boldsymbol{\beta})\to\boldsymbol{\beta}}$ vector appearing on the right hand side of the update expression refers to its value in the previous iteration.

Appendix C

This appendix contains additional details on algorithmic routines and derivations for implementing the algorithms described in Chapter 3. Some of them are adopted and/or inspired from Neville *et al.* (2014), Nolan and Wand (2020) and Nolan *et al.* (2020), to which we refer for further insights.

C.1 Multilevel Sparse Matrix Problems and Associated Routines

Algorithms 3.2 and 3.3 described in Chapter 3 rely upon two matrix algebraic routines for efficiently solving the two-level and three-level versions of the *multilevel sparse matrix problems* defined in Nolan and Wand (2020). They correspond to the SolveTwoLevelSparseMatrix and SolveThreeLevelSparseMatrix algorithms that we list hereafter as Algorithms C.1 and C.2, respectively.

We briefly describe two-level and three-level sparse matrix structures and give an explicit definition of the aforementioned routines for efficiently solving the associated linear system problems, following Appendix A of Nolan *et al.* (2020).

C.1.1 Two-Level Sparse Matrix Problems

Two-level sparse matrix problems are described in Section 2 of Nolan and Wand (2020). These problems are related to finding the vector $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{a}$, where

$$\boldsymbol{A} \equiv \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12,1} & \boldsymbol{A}_{12,2} & \cdots & \boldsymbol{A}_{12,m} \\ \boldsymbol{A}_{12,1}^T & \boldsymbol{A}_{22,1} & \boldsymbol{O} & \cdots & \boldsymbol{O} \\ \boldsymbol{A}_{12,2}^T & \boldsymbol{O} & \boldsymbol{A}_{22,2} & \cdots & \boldsymbol{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{12,m}^T & \boldsymbol{O} & \boldsymbol{O} & \cdots & \boldsymbol{A}_{22,m} \end{bmatrix}, \quad \boldsymbol{a} \equiv \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_{2,1} \\ \boldsymbol{a}_{2,2} \\ \vdots \\ \boldsymbol{a}_{2,m} \end{bmatrix} \quad \text{and} \quad \boldsymbol{x} \equiv \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_{2,1} \\ \boldsymbol{x}_{2,2} \\ \vdots \\ \boldsymbol{x}_{2,m} \end{bmatrix},$$

and obtaining the sub-blocks of $\boldsymbol{A}^{-1}$ corresponding to the non-zero blocks of $\boldsymbol{A}$. The structure of $\boldsymbol{A}^{-1}$ is

$$\boldsymbol{A}^{-1} \equiv \begin{bmatrix} \boldsymbol{A}^{11} & \boldsymbol{A}^{12,1} & \boldsymbol{A}^{12,2} & \cdots & \boldsymbol{A}^{12,m} \\ \boldsymbol{A}^{12,1\,T} & \boldsymbol{A}^{22,1} & \times & \cdots & \times \\ \boldsymbol{A}^{12,2\,T} & \times & \boldsymbol{A}^{22,2} & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{12,m} & \times & \times & \cdots & \boldsymbol{A}^{22,m} \end{bmatrix}.$$

The blocks represented by the $\times$ symbol are not of interest. Such problems are efficiently solved by the SolveTwoLevelSparseMatrix routine, which is here listed as Algorithm C.1 and is justified by Theorem 2.2 of Nolan and Wand (2020).

Algorithm C.1 *The* SolveTwoLevelSparseMatrix *algorithm for solving the two-level sparse matrix problem* $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{a}$ *and sub-blocks of* $\boldsymbol{A}^{-1}$ *corresponding to the non-zero sub-blocks of* $\boldsymbol{A}$.

Inputs: $\boldsymbol{a}_1(p \times 1), \boldsymbol{A}_{11}(p \times p), \left\{\left(\boldsymbol{a}_{2,i}(q \times 1), \boldsymbol{A}_{22,i}(q \times q), \boldsymbol{A}_{12,i}(p \times q)\right) : 1 \le i \le m\right\}$

$\boldsymbol{\omega} \longleftarrow \boldsymbol{a}_1 \quad ; \quad \boldsymbol{\Omega} \longleftarrow \boldsymbol{A}_{11}$

For $i = 1, \ldots, m$:

$$\boldsymbol{\omega} \longleftarrow \boldsymbol{\omega} - \boldsymbol{A}_{12,i}\boldsymbol{A}_{22,i}^{-1}\boldsymbol{a}_{2,i} \quad ; \quad \boldsymbol{\Omega} \longleftarrow \boldsymbol{\Omega} - \boldsymbol{A}_{12,i}\boldsymbol{A}_{22,i}^{-1}\boldsymbol{A}_{12,i}^{T}$$

$\boldsymbol{A}^{11} \longleftarrow \boldsymbol{\Omega}^{-1} \quad ; \quad \boldsymbol{x}_1 \longleftarrow \boldsymbol{A}^{11}\boldsymbol{\omega}$

For $i = 1, \ldots, m$:

$$\boldsymbol{x}_{2,i} \longleftarrow \boldsymbol{A}_{22,i}^{-1}(\boldsymbol{a}_{2,i} - \boldsymbol{A}_{12,i}^{T}\boldsymbol{x}_1) \quad ; \quad \boldsymbol{A}^{12,i} \longleftarrow -(\boldsymbol{A}_{22,i}^{-1}\boldsymbol{A}_{12,i}^{T}\boldsymbol{A}^{11})^{T}$$
$$\boldsymbol{A}^{22,i} \longleftarrow \boldsymbol{A}_{22,i}^{-1}(\boldsymbol{I} - \boldsymbol{A}_{12,i}^{T}\boldsymbol{A}^{12,i})$$

Outputs: $\boldsymbol{x}_1, \boldsymbol{A}^{11}, \left\{\left(\boldsymbol{x}_{2,i}, \boldsymbol{A}^{22,i}, \boldsymbol{A}^{12,i}\right) : 1 \le i \le m\right\}$.

C.1.2 Three-Level Sparse Matrix Problems

Three-level sparse matrix problems are described in Section 3 of Nolan and Wand (2020). An illustrative three-level sparse matrix example is

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12,1} & \boldsymbol{A}_{12,11} & \boldsymbol{A}_{12,12} & \boldsymbol{A}_{12,2} & \boldsymbol{A}_{12,21} & \boldsymbol{A}_{12,22} & \boldsymbol{A}_{12,23} \\ \boldsymbol{A}_{12,1}^T & \boldsymbol{A}_{22,1} & \boldsymbol{A}_{12,1,1} & \boldsymbol{A}_{12,1,2} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{A}_{12,11}^T & \boldsymbol{A}_{12,1,1}^T & \boldsymbol{A}_{22,11} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{A}_{12,12}^T & \boldsymbol{A}_{12,1,2}^T & \boldsymbol{O} & \boldsymbol{A}_{22,12} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{A}_{12,2}^T & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_{22,2} & \boldsymbol{A}_{12,2,1} & \boldsymbol{A}_{12,2,2} & \boldsymbol{A}_{12,2,3} \\ \boldsymbol{A}_{12,21}^T & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_{12,2,1}^T & \boldsymbol{A}_{22,21} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{A}_{12,22}^T & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_{12,2,2}^T & \boldsymbol{O} & \boldsymbol{A}_{22,22} & \boldsymbol{O} \\ \boldsymbol{A}_{12,23}^T & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_{12,2,3}^T & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_{22,23} \end{bmatrix},$$

which corresponds to level 2 group sizes $n_1 = 2$ and $n_2 = 3$, and a level 1 group size $m = 2$. A generic three-level sparse matrix $\boldsymbol{A}$ consists of the following components:

- A $p \times p$ matrix $\boldsymbol{A}_{11}$, which is assigned the $(1,1)$-block position.
- A set of partitioned matrices $\left\{ \left[\begin{array}{c|c|c|c} \boldsymbol{A}_{12,i} & \boldsymbol{A}_{12,ij} & \ldots & \boldsymbol{A}_{12,in_i} \end{array} \right] : 1 \le i \le m \right\}$, which is assigned the $(1,2)$-block position. For each $1 \le i \le m$, $\boldsymbol{A}_{12,i}$ is $p \times q_1$, and for each $1 \le j \le n_i$, $\boldsymbol{A}_{12,ij}$ is $p \times q_2$.
- A $(2,1)$-block, which is simply the transpose of the $(1,2)$-block.
- A block diagonal structure along the $(2,2)$-block position, where each sub-block is a two-level sparse matrix, as defined before. For each $1 \le i \le m$, $\boldsymbol{A}_{22,i}$ is $q_1 \times q_1$, and for each $1 \le j \le n_i$, $\boldsymbol{A}_{12,i,j}$ is $q_1 \times q_2$ and $\boldsymbol{A}_{22,ij}$ is $q_2 \times q_2$.

The three-level sparse matrix problem arising from the illustrative example above is defined as finding the vector $\boldsymbol{x}$ such that $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{a}$, where:

$$\boldsymbol{a} \equiv \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_{2,1} \\ \boldsymbol{a}_{2,11} \\ \boldsymbol{a}_{2,12} \\ \boldsymbol{a}_{2,2} \\ \boldsymbol{a}_{2,21} \\ \boldsymbol{a}_{2,22} \\ \boldsymbol{a}_{2,23} \end{bmatrix} \quad \text{and} \quad \boldsymbol{x} \equiv \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_{2,1} \\ \boldsymbol{x}_{2,11} \\ \boldsymbol{x}_{2,12} \\ \boldsymbol{x}_{2,2} \\ \boldsymbol{x}_{2,21} \\ \boldsymbol{x}_{2,22} \\ \boldsymbol{x}_{2,23} \end{bmatrix},$$

and determining the sub-blocks of $\boldsymbol{A}^{-1}$ corresponding to the non-zero sub-blocks of $\boldsymbol{A}$. The structure of $\boldsymbol{A}^{-1}$ is

$$\boldsymbol{A}^{-1} = \begin{bmatrix} \boldsymbol{A}^{11} & \boldsymbol{A}^{12,1} & \boldsymbol{A}^{12,11} & \boldsymbol{A}^{12,12} & \boldsymbol{A}^{12,2} & \boldsymbol{A}^{12,21} & \boldsymbol{A}^{12,22} & \boldsymbol{A}^{12,23} \\ \boldsymbol{A}^{12,1\,T} & \boldsymbol{A}^{22,1} & \boldsymbol{A}^{12,1,1} & \boldsymbol{A}^{12,1,2} & \times & \times & \times & \times \\ \boldsymbol{A}^{12,11\,T} & \boldsymbol{A}^{12,1,1\,T} & \boldsymbol{A}^{22,11} & \times & \times & \times & \times & \times \\ \boldsymbol{A}^{12,12\,T} & \boldsymbol{A}^{12,1,2\,T} & \times & \boldsymbol{A}^{22,12} & \times & \times & \times & \times \\ \boldsymbol{A}^{12,2\,T} & \times & \times & \times & \boldsymbol{A}^{22,2} & \boldsymbol{A}^{12,2,1} & \boldsymbol{A}^{12,2,2} & \boldsymbol{A}^{12,2,3} \\ \boldsymbol{A}^{12,21\,T} & \times & \times & \times & \boldsymbol{A}^{12,2,1\,T} & \boldsymbol{A}^{22,21} & \times & \times \\ \boldsymbol{A}^{12,22\,T} & \times & \times & \times & \boldsymbol{A}^{12,2,2\,T} & \times & \boldsymbol{A}^{22,22} & \times \\ \boldsymbol{A}^{12,23\,T} & \times & \times & \times & \boldsymbol{A}^{12,2,3\,T} & \times & \times & \boldsymbol{A}^{22,23} \end{bmatrix}.$$

For a general three-level sparse linear system problem, $\boldsymbol{a}_1$ and $\boldsymbol{x}_1$ are $p \times 1$ vectors. For each $1 \leq i \leq m$, $\boldsymbol{a}_{2,i}$ and $\boldsymbol{x}_{2,i}$ are $q_1 \times 1$ vectors. For each $1 \leq i \leq m$ and $1 \leq j \leq n_i$ the vectors $\boldsymbol{a}_{2,ij}$ and $\boldsymbol{x}_{2,ij}$ have dimension $q_2 \times 1$.

Such problems are efficiently solved by the SolveThreeLevelSparseMatrix routine, which is here listed as Algorithm C.2 and is justified by Theorem 3.2 of Nolan and Wand (2020).

Algorithm C.2 *The* SolveThreeLevelSparseMatrix *algorithm for solving the three-level sparse matrix problem* $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{a}$ *and sub-blocks of* $\boldsymbol{A}^{-1}$ *corresponding to the non-zero sub-blocks of* $\boldsymbol{A}$.

Input: $\boldsymbol{a}_1(p \times 1), \boldsymbol{A}_{11}(p \times p)$, $\big\{\big(\boldsymbol{a}_{2,i}(q_1 \times 1), \boldsymbol{A}_{22,i}(q_1 \times q_1), \boldsymbol{A}_{12,i}(p \times q_1) : 1 \le i \le m\big\}$, $\big\{\boldsymbol{a}_{2,ij}(q_2 \times 1), \boldsymbol{A}_{22,ij}(q_2 \times q_2), \boldsymbol{A}_{12,ij}(p \times q_2), \boldsymbol{A}_{12,i,j}(q_1 \times q_2)\big) : 1 \le i \le m,\ 1 \le j \le n_i\big\}$.

$\boldsymbol{\omega} \longleftarrow \boldsymbol{a}_1 \quad ; \quad \boldsymbol{\Omega} \longleftarrow \boldsymbol{A}_{11}$

For $i = 1, \ldots, m$:

$\boldsymbol{h}_{2,i} \longleftarrow \boldsymbol{a}_{2,i} \quad ; \quad \boldsymbol{H}_{12,i} \longleftarrow \boldsymbol{A}_{12,i} \quad ; \quad \boldsymbol{H}_{22,i} \longleftarrow \boldsymbol{A}_{22,i}$

For $j = 1, \ldots, n_i$:

$\boldsymbol{h}_{2,i} \longleftarrow \boldsymbol{h}_{2,i} - \boldsymbol{A}_{12,i,j}\boldsymbol{A}_{22,ij}^{-1}\boldsymbol{a}_{2,ij} \quad ; \quad \boldsymbol{H}_{12,i} \longleftarrow \boldsymbol{H}_{12,i} - \boldsymbol{A}_{12,ij}\boldsymbol{A}_{22,ij}^{-1}\boldsymbol{A}_{12,i,j}^{T}$

$\boldsymbol{H}_{22,i} \longleftarrow \boldsymbol{H}_{22,i} - \boldsymbol{A}_{12,i,j}\boldsymbol{A}_{22,ij}^{-1}\boldsymbol{A}_{12,i,j}^{T}$

$\boldsymbol{\omega} \longleftarrow \boldsymbol{\omega} - \boldsymbol{A}_{12,ij}\boldsymbol{A}_{22,ij}^{-1}\boldsymbol{a}_{2,ij} \quad ; \quad \boldsymbol{\Omega} \longleftarrow \boldsymbol{\Omega} - \boldsymbol{A}_{12,ij}\boldsymbol{A}_{22,ij}^{-1}\boldsymbol{A}_{12,ij}^{T}$

$\boldsymbol{\omega} \longleftarrow \boldsymbol{\omega} - \boldsymbol{H}_{12,i}\boldsymbol{H}_{22,i}^{-1}\boldsymbol{h}_{2,i} \quad ; \quad \boldsymbol{\Omega} \longleftarrow \boldsymbol{\Omega} - \boldsymbol{H}_{12,i}\boldsymbol{H}_{22,i}^{-1}\boldsymbol{H}_{12,i}^{T}$

$\boldsymbol{A}^{11} \longleftarrow \boldsymbol{\Omega}^{-1} \quad ; \quad \boldsymbol{x}_1 \longleftarrow \boldsymbol{A}^{11}\boldsymbol{\omega}$

For $i = 1, \ldots, m$:

$\boldsymbol{x}_{2,i} \longleftarrow \boldsymbol{H}_{22,i}^{-1}(\boldsymbol{h}_{2,i} - \boldsymbol{H}_{12,i}^{T}\boldsymbol{x}_1) \quad ; \quad \boldsymbol{A}^{12,i} \longleftarrow -(\boldsymbol{H}_{22,i}^{-1}\boldsymbol{H}_{12,i}^{T}\boldsymbol{A}^{11})^{T}$

$\boldsymbol{A}^{22,i} \longleftarrow \boldsymbol{H}_{22,i}^{-1}(\boldsymbol{I} - \boldsymbol{H}_{12,i}^{T}\boldsymbol{A}^{12,i})$

For $j = 1, \ldots, n_i$:

$\boldsymbol{x}_{2,ij} \longleftarrow \boldsymbol{A}_{22,ij}^{-1}\big(\boldsymbol{a}_{2,ij} - \boldsymbol{A}_{12,ij}^{T}\boldsymbol{x}_1 - \boldsymbol{A}_{12,i,j}^{T}\boldsymbol{x}_{2,i}\big)$

$\boldsymbol{A}^{12,ij} \longleftarrow -\big\{\boldsymbol{A}_{22,ij}^{-1}\big(\boldsymbol{A}_{12,ij}^{T}\boldsymbol{A}^{11} + \boldsymbol{A}_{12,i,j}^{T}\,\boldsymbol{A}^{12,i\,T}\big)\big\}^{T}$

$\boldsymbol{A}^{12,i,j} \longleftarrow -\big\{\boldsymbol{A}_{22,ij}^{-1}\big(\boldsymbol{A}_{12,ij}^{T}\,\boldsymbol{A}^{12,i} + \boldsymbol{A}_{12,i,j}^{T}\,\boldsymbol{A}^{22,i}\big)\big\}^{T}$

$\boldsymbol{A}^{22,ij} \longleftarrow \boldsymbol{A}_{22,ij}^{-1}\big(\boldsymbol{I} - \boldsymbol{A}_{12,ij}^{T}\boldsymbol{A}^{12,ij} - \boldsymbol{A}_{12,i,j}^{T}\,\boldsymbol{A}^{12,i,j}\big)$

Outputs: $\boldsymbol{x}_1, \boldsymbol{A}^{11}, \big\{\big(\boldsymbol{x}_{2,i}, \boldsymbol{A}^{22,i}, \boldsymbol{A}^{12,i}\big) : 1 \le i \le m\big\}$,
$\big\{\big(\boldsymbol{x}_{2,ij}, \boldsymbol{A}^{22,ij}, \boldsymbol{A}^{12,ij}, \boldsymbol{A}^{12,i,j},\big) : 1 \le i \le m,\ 1 \le j \le n_i\big\}$.

C.2 Derivations

We derive explicit updates for the parameters of the optimal density functions in (3.13) using arguments similar to those provided in Neville *et al.* (2014). Such updates are then combined with the derivations given in Appendix B of Nolan *et al.* (2020) for deriving the streamlined MFVB algorithms for our two- and three- level linear mixed-effects models, i.e. Algorithms 3.2 and 3.3.

C.2.1 Derivation of $\mathfrak{q}^*(\beta_0, \beta)$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for the parameter vector $(\beta_0, \boldsymbol{\beta})$ is:

$$\begin{aligned}\mathfrak{p}(\beta_0, \boldsymbol{\beta}|\text{rest}) &\propto \mathfrak{p}(\boldsymbol{y}|\beta_0, \boldsymbol{\beta}, \sigma^2)\, \mathfrak{p}(\beta_0)\, \mathfrak{p}(\boldsymbol{\beta}|\boldsymbol{\zeta}, \tau^2) \\ &\propto \exp\left\{ -\frac{1}{2}\left[\begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix}^T \left(\frac{1}{\sigma^2} \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix}^T \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix} + \begin{bmatrix} \sigma_{\beta_0}^{-2} & \mathbf{0}^T \\ \mathbf{0} & \tau^{-2} \operatorname{diag}(\boldsymbol{\zeta}) \end{bmatrix} \right) \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix} \right.\right. \\ &\qquad \left.\left. - 2 \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix}^T \left(\frac{1}{\sigma^2} \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix}^T \boldsymbol{y} + \begin{bmatrix} \sigma_{\beta_0}^{-2} & \mathbf{0}^T \\ \mathbf{0} & \tau^{-2} \operatorname{diag}(\boldsymbol{\zeta}) \end{bmatrix} \begin{bmatrix} \mu_{\beta_0} \\ \mathbf{0} \end{bmatrix} \right) \right] \right\}\end{aligned}$$

and application of (1.6) results in:

$$\begin{aligned}\mathfrak{q}^*(\beta_0, \boldsymbol{\beta}) &\propto \exp\left\{ E_{\mathfrak{q}}\{\log \mathfrak{p}(\beta_0, \boldsymbol{\beta}|\text{rest})\} \right\} \\ &\propto \exp\left\{ -\frac{1}{2}\left[\begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix}^T \left(E_{\mathfrak{q}}\{\sigma^{-2}\} \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix}^T \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix} + \begin{bmatrix} \sigma_{\beta_0}^{-2} & \mathbf{0}^T \\ \mathbf{0} & E_{\mathfrak{q}}\{\tau^{-2}\} \operatorname{diag}(E_{\mathfrak{q}}\{\boldsymbol{\zeta}\}) \end{bmatrix} \right) \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix} \right.\right. \\ &\qquad \left.\left. - 2 \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix}^T \left(E_{\mathfrak{q}}\{\sigma^{-2}\} \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix}^T \boldsymbol{y} + \begin{bmatrix} \sigma_{\beta_0}^{-2} & \mathbf{0}^T \\ \mathbf{0} & E_{\mathfrak{q}}\{\tau^{-2}\} \operatorname{diag}(E_{\mathfrak{q}}\{\boldsymbol{\zeta}\}) \end{bmatrix} \begin{bmatrix} \mu_{\beta_0} \\ \mathbf{0} \end{bmatrix} \right) \right] \right\}.\end{aligned}$$

After *completion of the square* manipulations for the multivariate Gaussian distribution and standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\beta_0, \boldsymbol{\beta}) \text{ is a } \mathrm{N}\left(\boldsymbol{\mu}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})}, \boldsymbol{\Sigma}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})}\right) \text{ density function}$$

with

$$\boldsymbol{\Sigma}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})} \longleftarrow \left(\mu_{\mathfrak{q}(1/\sigma^2)} \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix}^T \begin{bmatrix} \mathbf{1} \,|\, \boldsymbol{X} \end{bmatrix} + \begin{bmatrix} \sigma_{\beta_0}^{-2} & \mathbf{0}^T \\ \mathbf{0} & \mu_{\mathfrak{q}(1/\tau^2)} \operatorname{diag}(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\zeta})}) \end{bmatrix} \right)^{-1}$$

and

$$\boldsymbol{\mu}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})} \longleftarrow \boldsymbol{\Sigma}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})} \left(\mu_{\mathfrak{q}(1/\sigma^2)} \left[\mathbf{1} \mid \boldsymbol{X} \right]^T \boldsymbol{y} + \begin{bmatrix} \mu_{\beta_0}/\sigma^2_{\beta_0} \\ \mathbf{0} \end{bmatrix} \right).$$

C.2.2 Derivation of $\mathfrak{q}^*(\sigma^2)$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for parameter σ^2 is:

$$\begin{aligned} \mathfrak{p}(\sigma^2|\text{rest}) &\propto \mathfrak{p}(\boldsymbol{y}|\beta_0,\boldsymbol{\beta},\sigma^2)\,\mathfrak{p}(\sigma^2|a_{\sigma^2}) \\ &\propto (\sigma^2)^{-(\nu_{\sigma^2}+n)/2-1} \exp\left\{ -\frac{1}{\sigma^2} \frac{a_{\sigma^2}^{-1} + \|\boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta}\|^2}{2} \right\} \end{aligned}$$

and application of (1.6) results in:

$$\begin{aligned} \mathfrak{q}^*(\sigma^2) &\propto \exp\left\{ E_{\mathfrak{q}}\{\log \mathfrak{p}(\sigma^2|\text{rest})\} \right\} \\ &\propto (\sigma^2)^{-(\nu_{\sigma^2}+n)/2-1} \exp\left\{ -\frac{1}{\sigma^2} \frac{E_{\mathfrak{q}}\{a_{\sigma^2}^{-1}\} + E_{\mathfrak{q}}\{\|\boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta}\|^2\}}{2} \right\}. \end{aligned}$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\sigma^2) \text{ is an Inverse-}\chi^2\left(\xi_{\mathfrak{q}(\sigma^2)}, \lambda_{\mathfrak{q}(\sigma^2)}\right) \text{ density function}$$

with

$$\xi_{\mathfrak{q}(\sigma^2)} \longleftarrow n + \nu_{\sigma^2}$$

and

$$\lambda_{\mathfrak{q}(\sigma^2)} \longleftarrow \mu_{\mathfrak{q}(1/a_{\sigma^2})} + \left\| \boldsymbol{y} - \left[\mathbf{1} \mid \boldsymbol{X} \right] \boldsymbol{\mu}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})} \right\|^2 + \text{tr}\left\{ \boldsymbol{\Sigma}_{\mathfrak{q}(\beta_0,\boldsymbol{\beta})} \left[\mathbf{1} \mid \boldsymbol{X} \right]^T \left[\mathbf{1} \mid \boldsymbol{X} \right] \right\}.$$

C.2.3 Derivation of $\mathfrak{q}^*(a_{\sigma^2})$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for parameter a_{σ^2} is:

$$\begin{aligned} \mathfrak{p}(a_{\sigma^2}|\text{rest}) &\propto \mathfrak{p}(\sigma^2|a_{\sigma^2})\,\mathfrak{p}(a_{\sigma^2}) \\ &\propto (a_{\sigma^2})^{-(\nu_{\sigma^2}+1)/2-1} \exp\left\{ -\frac{1}{a_{\sigma^2}} \frac{\sigma^{-2} + (\nu_{\sigma^2} s^2_{\sigma^2})^{-1}}{2} \right\} \end{aligned}$$

and application of (1.6) results in:

$$\begin{aligned}\mathfrak{q}^*(a_{\sigma^2}) &\propto \exp\left\{E_{\mathfrak{q}}\{\log \mathfrak{p}(a_{\sigma^2}|\text{rest})\}\right\} \\ &\propto (a_{\sigma^2})^{-(\nu_{\sigma^2}+1)/2-1} \exp\left\{-\frac{1}{a_{\sigma^2}}\frac{E_{\mathfrak{q}}\{\sigma^{-2}\}+(\nu_{\sigma^2}s^2_{\sigma^2})^{-1}}{2}\right\}.\end{aligned}$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(a_{\sigma^2}) \text{ is an Inverse-}\chi^2\left(\xi_{\mathfrak{q}(a_{\sigma^2})}, \lambda_{\mathfrak{q}(a_{\sigma^2})}\right) \text{ density function}$$

with

$$\xi_{\mathfrak{q}(a_{\sigma^2})} \longleftarrow 1+\nu_{\sigma^2} \quad \text{and} \quad \lambda_{\mathfrak{q}(a_{\sigma^2})} \longleftarrow \mu_{\mathfrak{q}(1/\sigma^2)} + (\nu_{\sigma^2}s^2_{\sigma^2})^{-1}.$$

C.2.4 Derivation of $\mathfrak{q}^*(\tau^2)$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for parameter τ^2 is:

$$\begin{aligned}\mathfrak{p}(\tau^2|\text{rest}) &\propto \left\{\prod_{h=1}^{H}\mathfrak{p}(\beta_h|\tau^2)\right\}\mathfrak{p}(\tau^2|a_{\tau^2}) \\ &\propto (\tau^2)^{-(H+1)/2-1}\exp\left\{-\frac{1}{\tau^2}\frac{a_{\tau^2}^{-1}+\sum_{h=1}^{H}\zeta_h(\beta_h)^2}{2}\right\}\end{aligned}$$

and application of (1.6) results in:

$$\begin{aligned}\mathfrak{q}^*(\tau^2) &\propto \exp\left\{E_{\mathfrak{q}}\{\log \mathfrak{p}(\tau^2|\text{rest})\}\right\} \\ &\propto (\tau^2)^{-(H+1)/2-1}\exp\left\{-\frac{1}{\tau^2}\frac{E_{\mathfrak{q}}\{a_{\tau^2}^{-1}\}+\sum_{h=1}^{H}E_{\mathfrak{q}}\{\zeta_h\}E_{\mathfrak{q}}\{\beta_h^2\}}{2}\right\}.\end{aligned}$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\tau^2) \text{ is an Inverse-}\chi^2\left(\xi_{\mathfrak{q}(\tau^2)}, \lambda_{\mathfrak{q}(\tau^2)}\right) \text{ density function}$$

with

$$\xi_{\mathfrak{q}(\tau^2)} \longleftarrow H+1 \quad \text{and} \quad \lambda_{\mathfrak{q}(\tau^2)} \longleftarrow \mu_{\mathfrak{q}(1/a_{\tau^2})} + \boldsymbol{\mu}^T_{\mathfrak{q}(\boldsymbol{\zeta})}\boldsymbol{\mu}_{\mathfrak{q}((\boldsymbol{\beta})^2)}.$$

C.2.5 Derivation of $\mathfrak{q}^*(a_{\tau^2})$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for parameter a_{τ^2} is:

$$\begin{aligned}\mathfrak{p}(a_{\tau^2}|\text{rest}) &\propto \mathfrak{p}(\tau^2|a_{\tau^2})\mathfrak{p}(a_{\tau^2}) \\ &\propto (a_{\tau^2})^{-2}\exp\left\{-\frac{1}{a_{\tau^2}}\frac{\tau^{-2}+s_{\tau^2}^{-2}}{2}\right\}\end{aligned}$$

and application of (1.6) results in:

$$\mathfrak{q}^*(a_{\tau^2}) \propto \exp\left\{E_{\mathfrak{q}}\{\log \mathfrak{p}(a_{\tau^2}|\text{rest})\}\right\} \propto (a_{\tau^2})^{-2}\exp\left\{-\frac{1}{a_{\tau^2}}\frac{E_{\mathfrak{q}}\{\tau^{-2}\}+s_{\tau^2}^{-2}}{2}\right\}.$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(a_{\tau^2}) \text{ is an Inverse-}\chi^2\left(\xi_{\mathfrak{q}(a_{\tau^2})}, \lambda_{\mathfrak{q}(a_{\tau^2})}\right) \text{ density function}$$

with

$$\xi_{\mathfrak{q}(a_{\tau^2})} \longleftarrow 2 \quad \text{and} \quad \lambda_{\mathfrak{q}(a_{\tau^2})} \longleftarrow \mu_{\mathfrak{q}(1/\tau^2)} + s_{\tau^2}^{-2}.$$

C.2.6 Derivation of $\mathfrak{q}^*(\zeta_h)$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for parameters ζ_h, $1 \leq h \leq H$, is:

$$\mathfrak{p}(\zeta_h|\text{rest}) \propto \mathfrak{p}(\beta_h|\zeta_h, \tau)\, \mathfrak{p}(\zeta_h|a_{\zeta_h}),$$

where $\mathfrak{p}(\beta_h|\zeta_h)$ is the density function of a $\text{N}(0, \tau^2/\zeta_h)$ distribution, while $\mathfrak{p}(\zeta_h|a_{\zeta_h})$ depends on the hierarchical specification given in Table 3.1 for each of the considered global-local priors.

Laplace Global-Local Prior Specification

If $\beta_h|\tau \sim \text{Laplace}(0, \tau)$ then:

$$\mathfrak{p}(\zeta_h|\text{rest}) \propto (\zeta_h)^{-3/2}\exp\left\{-\zeta_h\frac{\beta_h^2}{2\tau^2} - \frac{1}{2\zeta_h}\right\}$$

and application of (1.6) results in:

$$\mathfrak{q}^*(\zeta_h) \propto \exp\left\{E_{\mathfrak{q}}\{\log \mathfrak{p}(\zeta_h|\text{rest})\}\right\} \propto (\zeta_h)^{-3/2} \exp\left\{-\zeta_h \frac{E_{\mathfrak{q}}\{\beta_h^2\} E_{\mathfrak{q}}\{\tau^{-2}\}}{2} - \frac{1}{2\zeta_h}\right\}.$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\zeta_h) \text{ is an Inverse-Gaussian}(\mu_{\mathfrak{q}(\zeta_h)}, 1) \text{ density function}$$

with

$$\mu_{\mathfrak{q}(\zeta_h)} \longleftarrow \sqrt{1 \Big/ \left(\mu_{\mathfrak{q}(1/\tau^2)} \mu_{\mathfrak{q}(\beta_h^2)}\right)}.$$

Horseshoe Global-Local Prior Specification

If $\beta_h|\tau \sim \text{Horseshoe}(0, \tau)$ then:

$$\mathfrak{p}(\zeta_h|\text{rest}) \propto \exp\left\{-\zeta_h \left(\frac{\beta_h^2}{2\tau^2} + a_{\zeta_h}\right)\right\}$$

and application of (1.6) results in:

$$\mathfrak{q}^*(\zeta_h) \propto \exp\{E_{\mathfrak{q}}\{\log \mathfrak{p}(\zeta_h|\text{rest})\}\} \propto \exp\left\{-\zeta_h \left(\frac{E_{\mathfrak{q}}\{\beta_h^2\} E_{\mathfrak{q}}\{\tau^{-2}\}}{2} + E_{\mathfrak{q}}\{a_{\zeta_h}\}\right)\right\}.$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\zeta_h) \text{ is a Gamma}(1, \lambda_{\mathfrak{q}(\zeta_h)}) \text{ density function}$$

with

$$\lambda_{\mathfrak{q}(\zeta_h)} \longleftarrow \frac{\mu_{\mathfrak{q}(1/\tau^2)} \mu_{\mathfrak{q}(\beta_h^2)}}{2} + \mu_{\mathfrak{q}(a_{\zeta_h})}.$$

Normal-Exponential-Gamma Global-Local Prior Specification

If $\beta_h|\tau \sim \text{NEG}(0, \tau, \lambda)$ then:

$$\mathfrak{p}(\zeta_h|\text{rest}) \propto (\zeta_h)^{-3/2} \exp\left\{-\zeta_h \frac{\beta_h^2}{2\tau^2} - \frac{a_{\zeta_h}}{\zeta_h}\right\}$$

and application of (1.6) results in:

$$\mathfrak{q}^*(\zeta_h) \propto \exp\{E_{\mathfrak{q}}\{\log \mathfrak{p}(\zeta_h|\text{rest})\}\} \propto (\zeta_h)^{-3/2} \exp\left\{-\zeta_h \frac{E_{\mathfrak{q}}\{\beta_h^2\} E_{\mathfrak{q}}\{\tau^{-2}\}}{2} - \frac{E_{\mathfrak{q}}\{a_{\zeta_h}\}}{\zeta_h}\right\}.$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\zeta_h) \text{ is an Inverse-Gaussian}(\mu_{\mathfrak{q}(\zeta_h)}, \lambda_{\mathfrak{q}(\zeta_h)}) \text{ density function}$$

with

$$\mu_{\mathfrak{q}(\zeta_h)} \longleftarrow \sqrt{2\mu_{\mathfrak{q}(a_{\zeta_h})} \Big/ \left(\mu_{\mathfrak{q}(1/\tau^2)} \mu_{\mathfrak{q}(\beta_h^2)}\right)} \quad \text{and} \quad \lambda_{\mathfrak{q}(\zeta_h)} \longleftarrow 2\mu_{\mathfrak{q}(a_{\zeta_h})}.$$

C.2.7 Derivation of $\mathfrak{q}^*(a_{\zeta_h})$ and Associated Parameter Updates

Given the model formulation (3.12), the full conditional density function for parameters ζ_h, $1 \leq h \leq H$, is:

$$\mathfrak{p}(a_{\zeta_h}|\text{rest}) \propto \mathfrak{p}(\zeta_h|a_{\zeta_h})\, \mathfrak{p}(a_{\zeta_h}),$$

where both $\mathfrak{p}(\zeta_h|a_{\zeta_h})$ and $\mathfrak{p}(a_{\zeta_h})$ depend on the hierarchical specification given in Table 3.1 for each of the considered global-local priors.

Laplace Global-Local Prior Specification

According to Table 3.1, for the Laplace prior it is not necessary to introduce the auxiliary variables a_{ζ_h}. Hence $\mathfrak{q}(a_{\zeta_h})$ is not present and does not need to be determined, for all $1 \leq h \leq H$.

Horseshoe Global-Local Prior Specification

If $\beta_h|\tau \sim \text{Horseshoe}(0, \tau)$ then:

$$\mathfrak{p}(a_{\zeta_h}|\text{rest}) \propto \exp\{-(\zeta_h + 1)a_{\zeta_h}\}$$

and application of (1.6) results in:

$$\mathfrak{q}^*(a_{\zeta_h}) \propto \exp\{E_{\mathfrak{q}}\{\log \mathfrak{p}(a_{\zeta_h}|\text{rest})\}\} \propto \exp\left\{-a_{\zeta_h}\left(E_{\mathfrak{q}}\{\zeta_h\} + 1\right)\right\}.$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(a_{\zeta_h}) \text{ is a Gamma}(1, \lambda_{\mathfrak{q}(a_{\zeta_h})}) \text{ density function}$$

with

$$\lambda_{\mathfrak{q}(\zeta_h)} \longleftarrow \mu_{\mathfrak{q}(\zeta_h)} + 1.$$

Normal-Exponential-Gamma Global-Local Prior Specification

If $\beta_h|\tau \sim \text{NEG}(0, \tau, \lambda)$ then:

$$\mathfrak{p}(a_{\zeta_h}|\text{rest}) \propto (a_{\zeta_h})^{(\lambda+1)-1} \exp\{a_{\zeta_h}(\zeta_h^{-1} + 1)\}$$

and application of (1.6) results in:

$$\mathfrak{q}^*(a_{\zeta_h}) \propto \exp\left\{E_{\mathfrak{q}}\{\log \mathfrak{p}(a_{\zeta_h}|\text{rest})\}\right\} \propto (a_{\zeta_h})^{(\lambda+1)-1} \exp\{a_{\zeta_h}(E_{\mathfrak{q}}\{\zeta_h^{-1}\} + 1)\}.$$

After standard algebraic manipulations, it follows immediately that:

$$\mathfrak{q}^*(\zeta_h) \text{ is a Gamma}(\lambda + 1, \lambda_{\mathfrak{q}(a_{\zeta_h})}) \text{ density function}$$

with

$$\lambda_{\mathfrak{q}(\zeta_h)} \longleftarrow \mu_{\mathfrak{q}(1/\zeta_h)} + 1.$$

Appendix D

This appendix contains additional details concerning the nonstationary HGP model (4.15) described in Chapter 4.

D.1 Derivation of the Posterior Predictive Distribution

Given the nonstationary GPR model (4.15), it is of interest to predict $\boldsymbol{f}_{\boldsymbol{X}^*} = (f(\boldsymbol{x}_1^*), \ldots, f(\boldsymbol{x}_{n^*}^*))^T$ over a set of n^* new predictors $\boldsymbol{X}^* = (\boldsymbol{x}_1^{*,T}, \ldots, \boldsymbol{x}_{n^*}^{*,T})^T$, exploiting the knowledge about the unknown regression function supplied by the observed data in $\mathcal{D}$. As such, we need to determine the posterior predictive distributions $\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}$ and $\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}$, following similar arguments given in Section 4.2 for the generic (stationary) GPR model.

Without loss of generality, let we assume for now that the model parameter vector $\boldsymbol{\lambda} = (\boldsymbol{u}_{\boldsymbol{X}}, \sigma^2, \tau_u^2, \ell_u^2, \tau_f^2)^T$ is fixed and entirely known in advance. We will later consider its Bayesian treatment. Due to its hierarchical construction, predicting from the nonstationary GPR model requires first obtaining the prediction for the unknown latent GP prior over $\boldsymbol{X}^*$ and then to employ it together with $\boldsymbol{u}_{\boldsymbol{X}}$ for predicting $\boldsymbol{f}_{\boldsymbol{X}^*}$.

Let's start from the top GP layer. Given the vectors $\boldsymbol{u}_{\boldsymbol{X}^*} = (u(\boldsymbol{x}_1^*), \ldots, u(\boldsymbol{x}_{n^*}^*))^T$, $\boldsymbol{m}_{u;\boldsymbol{X}^*} = (m_u(\boldsymbol{x}_1^*), \ldots, m_u(\boldsymbol{x}_{n^*}^*))^T$ and matrices $\boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}^*} = (k_{\text{S.E.}}(\boldsymbol{x}_i^*, \boldsymbol{x}_{i'}^*; \tau_u^2, \ell_u^2))_{1\leq i,i'\leq n^*}$, $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}^*} = (k_{\text{S.E.}}(\boldsymbol{x}_i, \boldsymbol{x}_{i'}^*; \tau_u^2, \ell_u^2))_{1\leq i\leq n, 1\leq i'\leq n^*}$ and $\boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}} = \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}^*}^T$, it is possible to express the joint density function for $\boldsymbol{u}_{\boldsymbol{X}}$ and $\boldsymbol{u}_{\boldsymbol{X}^*}$ as:

$$\begin{bmatrix} \boldsymbol{u}_{\boldsymbol{X}} \\ \boldsymbol{u}_{\boldsymbol{X}^*} \end{bmatrix} \sim \text{N}_{n+n^*}\left(\begin{bmatrix} \boldsymbol{m}_{u;\boldsymbol{X}} \\ \boldsymbol{m}_{u;\boldsymbol{X}^*} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}} & \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}^*} \\ \boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}} & \boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}^*} \end{bmatrix} \right).$$

The posterior predictive distribution for the top GP layer follows by applying standard conditioning formulas for the multivariate Gaussian distribution:

$$\begin{aligned} \boldsymbol{u}_{\boldsymbol{X}^*}|\boldsymbol{u}_{\boldsymbol{X}} \sim \text{N}_{n^*}\Big(& \boldsymbol{m}_{u;\boldsymbol{X}^*} + \boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}}\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}^{-1}(\boldsymbol{u}_{\boldsymbol{X}} - \boldsymbol{m}_{u;\boldsymbol{X}}), \\ & \boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}^*} - \boldsymbol{K}_{u;\boldsymbol{X}^*,\boldsymbol{X}}\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}^{-1}\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}^*}\Big). \end{aligned} \tag{D.1}$$

A commonly-suggested advice is to add a very small *nugget effect* on the diagonal of $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}$, preventing *ill-conditioning* problem. Therefore, samples from $\boldsymbol{u}_{\boldsymbol{X}^*}|\boldsymbol{u}_{\boldsymbol{X}}$ can be efficiently obtained substituting $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}$ with, e.g., $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}} + 10^{-4}\boldsymbol{I}$ in the above expression.

Now let define the vector $\boldsymbol{m}_{f;\boldsymbol{X}^*} = (m_f(\boldsymbol{x}_1^*, u_1^*), \ldots, m_f(\boldsymbol{x}_{n^*}^*, u_{n^*}^*))^T$ and matrices $\boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}^*} = (k_{\text{P.S.–S.E.}}(\boldsymbol{x}_i^*, \boldsymbol{x}_{i'}^*; u_i^*, u_{i'}^*, \tau_f^2))_{1\leq i,i'\leq n^*}$, $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}^*} = (k_{\text{P.S.–S.E.}}(\boldsymbol{x}_i, \boldsymbol{x}_{i'}^*; u_i, u_{i'}^*, \tau_f^2))_{1\leq i\leq n,1\leq i'\leq n^*}$ and $\boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}} = \boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}}^T$, where we explicitly indicated the unknown $u_i = (\boldsymbol{u}_{\boldsymbol{X}})_i$ $(1 \leq i \leq n)$ and $u_i^* = (\boldsymbol{u}_{\boldsymbol{X}^*})_i$ $(1 \leq i \leq n^*)$ upon which they are evaluated. With arguments similar to those given in Section 4.2, the joint density function of $\boldsymbol{y}$ and $\boldsymbol{f}_{\boldsymbol{X}^*}$, *conditional* on both $\boldsymbol{u}_{\boldsymbol{X}}$ and $\boldsymbol{u}_{\boldsymbol{X}^*}$, is expressible as:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_{\boldsymbol{X}^*} \end{bmatrix} \Big| \boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{u}_{\boldsymbol{X}^*} \sim \mathrm{N}_{n+n^*}\left(\begin{bmatrix} \boldsymbol{m}_{f;\boldsymbol{X}} \\ \boldsymbol{m}_{f;\boldsymbol{X}^*} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I} & \boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}^*} \\ \boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}} & \boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}^*} \end{bmatrix} \right).$$

Again, applying standard conditioning formulas for the multivariate Gaussian distribution it is immediate to obtain the posterior predictive distribution for the bottom GP layer, *conditional to both* $\boldsymbol{u}_{\boldsymbol{X}}$ *and* $\boldsymbol{u}_{\boldsymbol{X}^*}$, having expression:

$$\begin{aligned} \boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{u}_{\boldsymbol{X}^*} \sim \mathrm{N}_{n^*}\Big(& \boldsymbol{m}_{f;\boldsymbol{X}^*} + \boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}}(\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{m}_{f;\boldsymbol{X}}), \\ & \boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}^*} - \boldsymbol{K}_{f;\boldsymbol{X}^*,\boldsymbol{X}}(\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}^*}\Big). \end{aligned} \tag{D.2}$$

Combining both (D.1) and (D.2), the posterior predictive distribution for $\boldsymbol{f}_{\boldsymbol{X}^*}$, conditioned on $\boldsymbol{u}_{\boldsymbol{X}}$, is:

$$\mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{u}_{\boldsymbol{X}}) = \int \mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{u}_{\boldsymbol{X}^*})\, \mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}^*}|\boldsymbol{u}_{\boldsymbol{X}})\, \mathrm{d}\boldsymbol{u}_{\boldsymbol{X}^*}. \tag{D.3}$$

Nevertheless, an explicit closed-form expression is not available and it necessary to resort to Monte-Carlo methods for sampling from it.

If a fully-Bayesian inferential treatment is adopted for our nonstationary HGP model, as it is the case in Chapter 4, then each of the above expressions has to be rewritten also accounting for their conditional dependence upon σ^2, τ_u^2, ℓ_u^2 and τ_f^2. Consequently, (D.3) has to be rewritten as:

$$\mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{\lambda}) = \int \mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{u}_{\boldsymbol{X}^*}, \sigma^2, \tau_u^2)\, \mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}^*}|\boldsymbol{u}_{\boldsymbol{X}}, \tau_u^2, \ell_u^2)\, \mathrm{d}\boldsymbol{u}_{\boldsymbol{X}^*}$$

and finally the posterior predictive distribution is defined integrating out $\boldsymbol{\lambda}$ from the above expression with respect to its posterior distribution, yielding:

$$\mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}) = \iint \mathfrak{p}(\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{u}_{\boldsymbol{X}^*}, \sigma^2, \tau_u^2)\, \mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}^*}|\boldsymbol{u}_{\boldsymbol{X}}, \tau_u^2, \ell_u^2)\, \mathfrak{p}(\boldsymbol{\lambda}|\boldsymbol{y})\, \mathrm{d}\boldsymbol{u}_{\boldsymbol{X}^*}\mathrm{d}\boldsymbol{\lambda}. \tag{D.4}$$

If it is of interest to predict $\boldsymbol{y}_{\boldsymbol{X}^*}$ instead of $\boldsymbol{f}_{\boldsymbol{X}^*}$, the same derivation holds after substituting $\boldsymbol{K}_{f;\boldsymbol{X}^*\boldsymbol{X}^*} + \sigma^2\boldsymbol{I}$ to $\boldsymbol{K}_{f;\boldsymbol{X}^*\boldsymbol{X}^*}$ in (D.2). The resulting posterior predictive distribution is:

$$\mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}) = \iint \mathfrak{p}(\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}, \boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{u}_{\boldsymbol{X}^*}, \sigma^2, \tau_u^2)\, \mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}^*}|\boldsymbol{u}_{\boldsymbol{X}}, \tau_u^2, \ell_u^2)\, \mathfrak{p}(\boldsymbol{\lambda}|\boldsymbol{y})\, \mathrm{d}\boldsymbol{u}_{\boldsymbol{X}^*}\mathrm{d}\boldsymbol{\lambda}. \tag{D.5}$$

A prediction $\tilde{\boldsymbol{f}}_{\boldsymbol{X}^*}$ from (D.4) can be obtained via Monte-Carlo proceeding as follows:

1. Sample $\tilde{\boldsymbol{\lambda}} = (\tilde{\boldsymbol{u}}_{\boldsymbol{X}}, \tilde{\sigma^2}, \tilde{\tau}_u^2, \tilde{\ell}_u^2, \tilde{\tau}_f^2)^T$ from $\mathfrak{p}(\boldsymbol{\lambda}|\boldsymbol{y})$;
2. Sample $\tilde{\boldsymbol{u}}_{\boldsymbol{X}^*}$ from $\boldsymbol{u}_{\boldsymbol{X}^*}|\tilde{\boldsymbol{u}}_{\boldsymbol{X}}, \tilde{\tau}_u^2, \tilde{\ell}_u^2$ having distribution (D.1);
3. Sample $\tilde{\boldsymbol{f}}_{\boldsymbol{X}^*}$ from $\boldsymbol{f}_{\boldsymbol{X}^*}|\boldsymbol{y}, \tilde{\boldsymbol{u}}_{\boldsymbol{X}}, \tilde{\boldsymbol{u}}_{\boldsymbol{X}^*}, \tilde{\sigma}^2, \tilde{\tau}_f^2$ having distribution (D.2);
4. Interpret $\tilde{\boldsymbol{f}}_{\boldsymbol{X}^*}$ as a draw from (D.4).

Similarly, a prediction $\tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}$ from (D.5) follows the same step but replaces Step 3 sampling $\tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}$ from $\boldsymbol{y}_{\boldsymbol{X}^*}|\boldsymbol{y}, \tilde{\boldsymbol{u}}_{\boldsymbol{X}}, \tilde{\boldsymbol{u}}_{\boldsymbol{X}^*}, \tilde{\sigma}^2, \tilde{\tau}_f^2$ having distribution (D.2) with $\boldsymbol{K}_{f;\boldsymbol{X}^*\boldsymbol{X}^*}$ replaced by $\boldsymbol{K}_{f;\boldsymbol{X}^*\boldsymbol{X}^*} + \sigma^2\boldsymbol{I}$. Hence $\tilde{\boldsymbol{y}}_{\boldsymbol{X}^*}$ is interpretable as being drawn from (D.5).

D.2 Derivation of Algorithm 4.1 and Implementation Details

Algorithm 4.1 described in Chapter 4 performs GVA on the nonstationary GPR model (4.15) employing the factor parametrization for the covariance matrix proposed by Ong *et al.* (2018). As such, its algorithmic structure follows from the stochastic gradient ascent algorithm proposed in Section 3 of Ong *et al.* (2018) and summarized in their Algorithm 1, with minor modifications we describe hereafter.

Given the reparameterized parameter vector $\boldsymbol{\theta} = (\boldsymbol{u}_{\boldsymbol{X}}^T, \log\sigma^2, \log\tau_u^2, \log\ell_u^2, \log\tau_f^2)^T$ of dimension $(n+4)\times 1$, we are interested in approximating the posterior density function $\mathfrak{p}(\boldsymbol{\theta}|\boldsymbol{y})$ having expression (4.17) with a probability density function $\mathfrak{q}(\boldsymbol{\theta})$ expressible as (4.18). Following the stochastic gradient ascent-based method for solving GVA, summarized in Chapter 1, here $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})} = (\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}^T, \mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})})^T, \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}^T)^T$ denotes the parameter vector of $\mathfrak{q}(\boldsymbol{\theta})$ and GVA finds the optimal $\boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}^*$ for which the

lower-bound

$$\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})) = \log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta}) - \log \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})$$

is maximal.

The stochastic gradient ascent method employing the reparametrization trick is based upon the iterative computation of (1.24). Given that

$$z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}) = \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} + (\boldsymbol{\omega}^T \otimes \boldsymbol{I})\mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}) + \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} \circ \boldsymbol{\varrho},$$

we can compute

$$\frac{\mathrm{d}\, z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})}{\mathrm{d}\, \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}} = \left[\frac{\mathrm{d}\, z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})}{\mathrm{d}\, \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}} \quad \frac{\mathrm{d}\, z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})}{\mathrm{d}\, \mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})})} \quad \frac{\mathrm{d}\, z(\boldsymbol{\zeta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})})}{\mathrm{d}\, \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}} \right]$$

explicitly, since

$$\frac{\mathrm{d}\, \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} + (\boldsymbol{\omega}^T \otimes \boldsymbol{I})\mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}) + \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} \circ \boldsymbol{\varrho}}{\mathrm{d}\, \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}} = \boldsymbol{I}$$

$$\frac{\mathrm{d}\, \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} + (\boldsymbol{\omega}^T \otimes \boldsymbol{I})\mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}) + \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} \circ \boldsymbol{\varrho}}{\mathrm{d}\, \mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})})} = \boldsymbol{\omega}^T \otimes \boldsymbol{I}$$

$$\text{and} \quad \frac{\mathrm{d}\, \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})} + (\boldsymbol{\omega}^T \otimes \boldsymbol{I})\mathrm{vec}(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}) + \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})} \circ \boldsymbol{\varrho}}{\mathrm{d}\, \boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})}} = \mathrm{diag}(\boldsymbol{\varrho}).$$

Moreover, the first addendum of $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))$ reads:

$$\begin{aligned}
\log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta}) &= \log \mathfrak{p}(\boldsymbol{y}|\boldsymbol{\theta}) + \log \mathfrak{p}(\boldsymbol{\theta}) \\
&= \log \mathfrak{p}(\boldsymbol{y}|\boldsymbol{u}_{\boldsymbol{X}}, \log \sigma^2, \log \tau_f^2) + \log \mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}}|\log \tau_u^2, \log \ell_u^2) \\
&\quad + \log \mathfrak{p}(\log \sigma^2) + \log \mathfrak{p}(\log \tau_u^2) + \log \mathfrak{p}(\log \ell_u^2) + \log \mathfrak{p}(\log \tau_f^2),
\end{aligned}$$

where:

- $\mathfrak{p}(\boldsymbol{y}|\boldsymbol{u}_{\boldsymbol{X}}, \log \sigma^2, \log \tau_f^2)$ is the density function of a $\mathrm{N}_n(\boldsymbol{0}, \boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}} + \exp(\log \sigma^2)\boldsymbol{I})$ distribution, $\boldsymbol{u}_{\boldsymbol{X}}$ and $\log \tau_f^2$ taking their contribution into the Paciorek-Schervish covariance function specification for building $\boldsymbol{K}_{f;\boldsymbol{X},\boldsymbol{X}}$;
- $\mathfrak{p}(\boldsymbol{u}_{\boldsymbol{X}}|\log \tau_f^2, \log \ell_f^2)$ is the density function of a $\mathrm{N}_n(\boldsymbol{0}, \boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}})$ density function, $\log \tau_f^2$ and $\log \ell_f^2$ taking their contribution into the squared exponential covariance function specification for building $\boldsymbol{K}_{u;\boldsymbol{X},\boldsymbol{X}}$;

- $\mathfrak{p}(\log \sigma^2)$ is the density function of the random variable $\log(\sigma^2)$ built on $\sigma^2 \sim$ Inverse-Gamma$(\xi_{\sigma^2}, \lambda_{\sigma^2})$;
- similarly for the density functions $\mathfrak{p}(\log \tau_u^2)$, $\mathfrak{p}(\log \ell_u^2)$ and $\mathfrak{p}(\log \tau_f^2)$ following the prior specifications given in (4.15).

An explicit expression for $\nabla_{\boldsymbol{\theta}} \log \mathfrak{p}(\boldsymbol{y}, \boldsymbol{\theta})$ in (1.24) is generally not available for the whole parameter vector $\boldsymbol{\theta}$, and we resort to automatic differentiation techniques (Baydin *et al.*, 2017) for computing it.

The second addendum of $\log \underline{\mathfrak{p}}(\boldsymbol{y}; \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}))$ is simply the log-probability density function of $\mathfrak{q}(\boldsymbol{\theta})$, which is a $\mathrm{N}(\boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}; \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T + \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^2)$ density function in $\boldsymbol{\theta}$. Therefore,

$$\nabla_{\boldsymbol{\theta}} \log \mathfrak{q}(\boldsymbol{\theta}; \boldsymbol{\eta}_{\mathfrak{q}(\boldsymbol{\theta})}) = -(\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T + \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^2)^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_{\mathfrak{q}(\boldsymbol{\theta})}).$$

Notice Step 5 of Algorithm 4.1 requires the inverse for $\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})}$. It can be computed efficiently in terms of both memory and computation time employing the Woodbury identity:

$$(\boldsymbol{\Sigma}_{\mathfrak{q}(\boldsymbol{\theta})})^{-1} = (\boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T + \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^2)^{-1}$$
$$= \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^{-2} - \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^{-2} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})} (\boldsymbol{I} + \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^{-2} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})})^{-1} \boldsymbol{B}_{\mathfrak{q}(\boldsymbol{\theta})}^T \mathrm{diag}(\boldsymbol{d}_{\mathfrak{q}(\boldsymbol{\theta})})^{-2}.$$

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016) TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pp. 265–283. USA: USENIX Association.

Albert, J. H. and Chib, S. (1993) Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* **88**(422), 669–679.

Ali, S. M. and Silvey, S. D. (1966) A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **28**, 131–142.

Allaire, J. and Tang, Y. (2020) *tensorflow: R Interface to TensorFlow*. R package version 2.2.0. `https://CRAN.R-project.org/package=tensorflow`.

Amari, S.-i. (1985) *Differential-Geometrical Methods in Statistics*. Volume 28 of *Lecture Notes in Statistics*. Springer-Verlag, New York.

Amari, S.-I. (2009) α-divergence is unique, belonging to both f-divergence and Bregman divergence classes. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory* **55**(11), 4925–4931.

Andrews, D. F. and Mallows, C. L. (1974) Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **36**, 99–102.

Archambeau, C., Cornford, D., Opper, M. and Shawe-Taylor, J. (2007) Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, volume 1 of *Proceedings of Machine Learning Research*, pp. 1–16. Bletchley Park, UK: PMLR.

Armagan, A., Clyde, M. and Dunson, D. B. (2011) Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems*, volume 24, pp. 523–531. Curran Associates, Inc.

Armagan, A. and Dunson, D. B. (2011) Sparse variational analysis of linear mixed models for large data sets. *Statistics & Probability Letters* **81**(8), 1056–1062.

Armagan, A., Dunson, D. B. and Lee, J. (2013) Generalized double Pareto shrinkage. *Statistica Sinica* **23**(1), 119–143.

Atay-Kayis, A. and Massam, H. (2005) A Monte Carlo method for computing the marginal likelihood in nondecomposable Gaussian graphical models. *Biometrika* **92**(2), 317–335.

Ba, S. and Joseph, V. R. (2012) Composite Gaussian process models for emulating expensive functions. *The Annals of Applied Statistics* **6**(4), 1838–1860.

Baltagi, B. H. (2021) *Econometric Analysis of Panel Data*. Springer.

Banerjee, A., Dunson, D. B. and Tokdar, S. T. (2013) Efficient Gaussian process regression for large datasets. *Biometrika* **100**(1), 75–89.

Barber, D. and Bishop, C. (1998) Ensemble learning in Bayesian neural networks. In *Generalization in Neural Networks and Machine Learning*, pp. 215–237. Springer Verlag.

Barbieri, M. M. and Berger, J. O. (2004) Optimal predictive model selection. *The Annals of Statistics* **32**(3), 870–897.

Barndorff-Nielsen, O. (1978) *Information and Exponential Families in Statistical Theory*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Ltd., Chichester.

Barthelmé, S. and Chopin, N. (2014) Expectation propagation for likelihood-free inference. *Journal of the American Statistical Association* **109**(505), 315–333.

Bartholomew, D., Knott, M. and Moustaki, I. (2011) *Latent Variable Models and Factor Analysis*. Third edition. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., Chichester.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A. and Siskind, J. M. (2017) Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research (JMLR)* **18**, Paper No. 153, 43.

Beal, M. J. and Ghahramani, Z. (2003) The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In *Bayesian Statistics, 7*, pp. 453–463. Oxford Univ. Press, New York.

Bhadra, A., Datta, J., Polson, N. G. and Willard, B. (2017) The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis* **12**(4), 1105–1131.

Bhadra, A., Datta, J., Polson, N. G. and Willard, B. (2019) Lasso meets horseshoe: a survey. *Statistical Science* **34**(3), 405–427.

Bhattacharya, A., Chakraborty, A. and Mallick, B. K. (2016) Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika* **103**(4), 985–991.

Bhattacharya, A., Pati, D., Pillai, N. S. and Dunson, D. B. (2015) Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association* **110**(512), 1479–1490.

Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York.

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017) Variational inference: a review for statisticians. *Journal of the American Statistical Association* **112**(518), 859–877.

Bondell, H. D. and Reich, B. J. (2012) Consistent high-dimensional Bayesian variable selection via penalized credible regions. *Journal of the American Statistical Association* **107**(500), 1610–1624.

Bottou, L. (2010) Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Heidelberg: Physica-Verlag HD.

Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*. Cambridge University Press, Cambridge.

Bregman, L. (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics* **7**(3), 200–217.

Brown, H. and Prescott, R. (2015) *Applied Mixed Models in Medicine*. John Wiley & Sons.

Brown, L. D. (1986) *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Volume 9 of *Institute of Mathematical Statistics Lecture Notes—Monograph Series*. Institute of Mathematical Statistics, Hayward, CA.

Bui, T. D., Hernandez-Lobato, D., Hernandez-Lobato, J., Li, Y. and Turner, R. (2016) Deep Gaussian processes for regression using approximate expectation propagation. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1472–1481. New York, New York, USA: PMLR.

Bui, T. D., Yan, J. and Turner, R. E. (2017) A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research (JMLR)* **18**, Paper No. 104, 72.

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

Carbonetto, P. and Stephens, M. (2012) Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Analysis* **7**(1), 73–107.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. and Riddell, A. (2017) Stan: A probabilistic programming language. *Journal of Statistical Software* **76**(1), 1–32.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2009) Handling sparsity via the horseshoe. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 73–80. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2010) The horseshoe estimator for sparse signals. *Biometrika* **97**(2), 465–480.

Casella, G. and George, E. I. (1992) Explaining the Gibbs sampler. *The American Statistician* **46**(3), 167–174.

Castillo, I. (2008) Lower bounds for posterior rates with Gaussian process priors. *Electronic Journal of Statistics* **2**, 1281–1299.

Celisse, A., Daudin, J.-J. and Pierre, L. (2012) Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics* **6**, 1847–1899.

Challis, E. and Barber, D. (2013) Gaussian Kullback-Leibler approximate inference. *Journal of Machine Learning Research (JMLR)* **14**, 2239–2286.

Chen, W. Y. and Wand, M. P. (2020) Factor graph fragmentization of expectation propagation. *Journal of the Korean Statistical Society* **49**(3), 722–756.

Chen, Z. and Dunson, D. B. (2003) Random effects selection in linear mixed models. *Biometrics. Journal of the International Biometric Society* **59**(4), 762–769.

Choi, T. and Schervish, M. J. (2007) On posterior consistency in nonparametric regression problems. *Journal of Multivariate Analysis* **98**(10), 1969–1987.

Cichocki, A. and Amari, S.-i. (2010) Families of alpha- beta- and gamma-divergences: flexible and robust measures of similarities. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **12**(6), 1532–1568.

Consonni, G. and Marin, J.-M. (2008) Mean-field variational approximate Bayesian inference for latent variable models. *Computational Statistics & Data Analysis* **52**(2), 790–798.

Cressie, N. A. C. (1993) *Statistics for Spatial Data*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., New York.

Csató, L. and Opper, M. (2002) Sparse on-line Gaussian processes. *Neural Computation* **14**(3), 641–668.

Damianou, A. and Lawrence, N. D. (2013) Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 207–215. Scottsdale, Arizona, USA: PMLR.

Dehaene, G. and Barthelmé, S. (2015) Bounding errors of expectation-propagation. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 244–252. Cambridge, MA, USA: MIT Press.

Dehaene, G. and Barthelmé, S. (2018) Expectation propagation in the large data limit. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **80**(1), 199–217.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **39**(1), 1–38.

Devroye, L. and Györfi, L. (1985) *Nonparametric Density Estimation*. Wiley Series in Probability and Mathematical Statistics: Tracts on Probability and Statistics. John Wiley & Sons, Inc., New York.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J. and Blei, D. (2017) Variational inference via χ upper bound minimization. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Dunlop, M. M., Girolami, M. A., Stuart, A. M. and Teckentrup, A. L. (2018) How deep are deep Gaussian processes? *Journal of Machine Learning Research (JMLR)* **19**, Paper No. 54, 46.

Duvenaud, D., Rippel, O., Adams, R. and Ghahramani, Z. (2014) Avoiding pathologies in very deep networks. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 202–210. Reykjavik, Iceland: PMLR.

Eddelbuettel, D. (2013) *Seamless R and C++ Integration with Rcpp*. New York: Springer.

Eddelbuettel, D. and Balamuta, J. J. (2018) Extending R with C++: A brief introduction to Rcpp. *The American Statistician* **72**(1), 28–36.

Eddelbuettel, D. and Sanderson, C. (2014) RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics and Data Analysis* **71**, 1054–1063.

Efron, B. (1978) The geometry of exponential families. *The Annals of Statistics* **6**(2), 362–376.

Efron, B. and Hastie, T. (2016) *Computer Age Statistical Inference*. Volume 5 of *Institute of Mathematical Statistics (IMS) Monographs*. Cambridge University Press, New York.

Faes, C., Ormerod, J. T. and Wand, M. P. (2011) Variational Bayesian inference for parametric and nonparametric regression with missing data. *Journal of the American Statistical Association* **106**(495), 959–971.

Fan, Y. and Li, R. (2012) Variable selection in linear mixed effects models. *The Annals of Statistics* **40**(4), 2043–2068.

Fitzmaurice, G., Davidian, M., Verbeke, G. and Molenberghs, G. (2008) *Longitudinal Data Analysis*. CRC Press.

Frank, I. E. and Friedman, J. H. (1993) A statistical view of some chemometrics regression tools. *Technometrics* **35**(2), 109–135.

Frey, B. J. (2002) Extending factor graphs so as to unify directed and undirected graphical models. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, pp. 257–264.

Frey, B. J. and MacKay, D. J. (1998) A revolution: Belief propagation in graphs with cycles. *Advances in Neural Information Processing Systems* pp. 479–485.

Gehre, M., Jin, B. and Lu, X. (2014) An analysis of finite element approximation in electrical impedance tomography. *Inverse Problems* **30**(4), 1–24.

Gelfand, A. E., Sahu, S. K. and Carlin, B. P. (1995) Efficient parameterisations for normal linear mixed models. *Biometrika* **82**(3), 479–488.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85**(410), 398–409.

Gelfand, I. M. and Fomin, S. V. (1963) *Calculus of Variations*. Prentice-Hall, Inc., Englewood Cliffs, N.J.

Gelman, A. (2006) Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis* **1**(3), 515–533.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014) *Bayesian Data Analysis*. Third edition. Texts in Statistical Science Series. CRC Press, Boca Raton, FL.

Geweke, J. and Zhou, G. (1996) Measuring the Pricing Error of the Arbitrage Pricing Theory. *The Review of Financial Studies* **9**(2), 557–587.

Gibbs, M. N. (1997) *Bayesian Gaussian Processes for Regression and Classification*. Ph.D. thesis. University of Cambridge.

Girolami, M. and Rogers, S. (2006) Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation* **18**(8), 1790–1817.

Goldstein, H. (2010) *Multilevel Statistical Models*. John Wiley & Sons Inc.

Gramacy, R. B. and Lee, H. K. H. (2008) Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* **103**(483), 1119–1130.

Griffin, J. E. and Brown, P. J. (2010) Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis* **5**(1), 171–188.

Griffin, J. E. and Brown, P. J. (2011) Bayesian hyper-lassos with non-convex penalization. *Australian & New Zealand Journal of Statistics* **53**(4), 423–442.

Groll, A. and Tutz, G. (2014) Variable selection for generalized linear mixed models by L_1-penalized estimation. *Statistics and Computing* **24**(2), 137–154.

Hahn, P. R. and Carvalho, C. M. (2015) Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *Journal of the American Statistical Association* **110**(509), 435–448.

Hall, P. (1987) On Kullback-Leibler loss and density estimation. *The Annals of Statistics* **15**(4), 1491–1519.

Hall, P., Humphreys, K. and Titterington, D. M. (2002) On the adequacy of variational lower bound functions for likelihood-based inference in Markovian models with missing values. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **64**(3), 549–564.

Hall, P., Johnstone, I. M., Ormerod, J. T., Wand, M. P. and Yu, J. C. F. (2020) Fast and accurate binary response mixed model analysis via expectation propagation. *Journal of the American Statistical Association* **115**(532), 1902–1916.

Hall, P., Ormerod, J. T. and Wand, M. P. (2011a) Theory of Gaussian variational approximation for a Poisson mixed model. *Statistica Sinica* **21**(1), 369–389.

Hall, P., Pham, T., Wand, M. P. and Wang, S. S. J. (2011b) Asymptotic normality and valid inference for Gaussian variational approximation. *The Annals of Statistics* **39**(5), 2502–2532.

Han, S., Liao, X., Dunson, D. B. and Carin, L. (2016) Variational Gaussian copula inference. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 829–838. Cadiz, Spain: PMLR.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1), 97–109.

Heaton, M. J., Datta, A., Finley, A. O. and et al. (2019) A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological, and Environmental Statistics* **24**(3), 398–425.

Hebbal, A., Brevault, L., Balesdent, M., Taibi, E.-G. and Melab, N. (2018) Efficient global optimization using deep Gaussian processes. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8.

Heinonen, M., Mannerström, H., Rousu, J., Kaski, S. and Lähdesmäki, H. (2016) Non-stationary Gaussian process regression with hamiltonian monte carlo. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 732–740. Cadiz, Spain: PMLR.

Hernandez-Lobato, J., Li, Y., Rowland, M., Bui, T. D., Hernandez-Lobato, D. and Turner, R. (2016) Black-box alpha divergence minimization. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1511–1520. New York, New York, USA: PMLR.

Higdon, D., Swall, J. and Kern, J. (1999) Non-stationary spatial modeling. In *Bayesian Statistics, 6*, pp. 761–768. Oxford Univ. Press, New York.

Hinton, G. E. and van Camp, D. (1993) Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pp. 5–13. New York, NY, USA: Association for Computing Machinery.

Hinton, G. E., Osindero, S. and Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural Computation* **18**(7), 1527–1554.

Hoerl, A. E. and Kennard, R. W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**(1), 55–67.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013) Stochastic variational inference. *Journal of Machine Learning Research (JMLR)* **14**, 1303–1347.

Hoffman, M. D. and Gelman, A. (2014) The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research (JMLR)* **15**, 1593–1623.

Honkela, A., Tornio, M., Raiko, T. and Karhunen, J. (2008) Natural conjugate gradient in variational inference. In *Neural Information Processing*, pp. 305–314. Berlin, Heidelberg: Springer Berlin Heidelberg.

Huang, A. and Wand, M. P. (2013) Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis* **8**(2), 439–451.

Huggins, J., Kasprzak, M., Campbell, T. and Broderick, T. (2020) Validated variational inference via practical posterior error bounds. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1792–1802. PMLR.

Hughes, D. M., García-Fiñana, M. and Wand, M. P. (2021) Fast approximate inference for multivariate longitudinal data. *Biostatistics* (Year and page numbers pending.).

Hui, F. K. C., Müller, S. and Welsh, A. H. (2017) Joint selection in mixed models using regularized PQL. *Journal of the American Statistical Association* **112**(519), 1323–1333.

Jaakkola, T. S. and Jordan, M. I. (2000) Bayesian parameter estimation via variational methods. *Statistics and Computing* **10**(1), 25–37.

Jordan, M. I. (2004) Graphical models. *Statistical Science* **19**(1), 140–155.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. and Saul, L. K. (1999) An introduction to variational methods for graphical models. *Machine Learning* **37**, 183–233.

Journel, A. G. and Huijbregts, C. J. (1978) *Mining Geostatistics*. Academic Press.

Jylänki, P., Nummenmaa, A. and Vehtari, A. (2014) Expectation propagation for neural networks with sparsity-promoting priors. *Journal of Machine Learning Research (JMLR)* **15**, 1849–1901.

Jylänki, P., Vanhatalo, J. and Vehtari, A. (2011) Robust Gaussian process regression with a Student-*t* likelihood. *Journal of Machine Learning Research (JMLR)* **12**, 3227–3257.

Kammann, E. E. and Wand, M. P. (2003) Geoadditive models. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **52**.

Kim, A. S. I. and Wand, M. P. (2016) The explicit form of expectation propagation for a simple statistical model. *Electronic Journal of Statistics* **10**(1), 550–581.

Kim, A. S. I. and Wand, M. P. (2018) On expectation propagation for generalised, linear and mixed models. *Australian & New Zealand Journal of Statistics* **60**(1), 75–102.

Kingma, D. P. and Welling, M. (2014) Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kinney, S. K. and Dunson, D. B. (2007) Fixed and random effects selection in linear and logistic models. *Biometrics. Journal of the International Biometric Society* **63**(3), 690–698.

Klebanoff, M. A. (2009) The collaborative perinatal project: a 50-year retrospective. *Paediatric and Perinatal Epidemiology* **23**(1), 2–8.

Knoblauch, J., Jewson, J. and Damoulas, T. (2019) Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063* .

Knowles, D. and Minka, T. (2011) Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Koepernik, P. and Pfaf, F. (2021) Consistency of Gaussian process regression in metric spaces. *Journal of Machine Learning Research (JMLR)* **22**, Paper No. 244, 27.

Korte, A., Vilhjálmsson, B. J., Segura, V., Platt, A., Long, Q. and Nordborg, M. (2012) A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nature Genetics* **44**(9), 1066–1071.

Kschischang, F., Frey, B. and Loeliger, H.-A. (2001) Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* **47**(2), 498–519.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. and Blei, D. M. (2017) Automatic differentiation variational inference. *Journal of Machine Learning Research (JMLR)* **18**, Paper No. 14, 45.

Kullback, S. and Leibler, R. A. (1951) On information and sufficiency. *Annals of Mathematical Statistics* **22**, 79–86.

Kuss, M. and Rasmussen, C. (2004) Gaussian processes in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press.

Lalchand, V. and Rasmussen, C. E. (2020) Approximate inference for fully Bayesian Gaussian process regression. In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118 of *Proceedings of Machine Learning Research*, pp. 1–12. PMLR.

Lee, C. Y. Y. and Wand, M. P. (2016) Streamlined mean field variational Bayes for longitudinal and multilevel data analysis. *Biometrical Journal* **58**(4), 868–895.

Letac, G. and Massam, H. (2007) Wishart distributions for decomposable graphs. *The Annals of Statistics* **35**(3), 1278–1323.

Li, H. and Pati, D. (2017) Variable selection using shrinkage priors. *Computational Statistics & Data Analysis* **107**, 107–119.

Li, J., Wang, Z., Li, R. and Wu, R. (2015a) Bayesian group Lasso for nonparametric varying-coefficient models with application to functional genome-wide association studies. *The Annals of Applied Statistics* **9**(2), 640–664.

Li, Y., Hernández-Lobato, J. M. and Turner, R. E. (2015b) Stochastic expectation propagation. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Li, Y. and Turner, R. E. (2016) Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Li, Y., Wang, S., Song, P. X.-K., Wang, N., Zhou, L. and Zhu, J. (2018) Doubly regularized estimation and selection in linear mixed-effects models for high-dimensional longitudinal data. *Statistics and its Interface* **11**(4), 721–737.

Lindner, C. C. and Rodger, C. A. (2009) *Design Theory*. Second edition. Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL.

Liu, Q. and Pierce, D. A. (1994) A note on Gauss-Hermite quadrature. *Biometrika* **81**(3), 624–629.

Liu, Q. and Wang, D. (2016) Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Luenberger, D. G. and Ye, Y. (2008) *Linear and Nonlinear Programming*. Third edition, volume 116 of *International Series in Operations Research & Management Science*. Springer, New York.

Lunn, D. J., Thomas, A., Best, N. and Spiegelhalter, D. (2000) WinBUGS – A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing* **10**, 325–337.

Luts, J., Broderick, T. and Wand, M. P. (2014) Real-time semiparametric regression. *Journal of Computational and Graphical Statistics* **23**(3), 589–615.

Luts, J. and Wand, M. P. (2015) Variational inference for count response semiparametric regression. *Bayesian Analysis* **10**(4), 991–1023.

MacKay, D. J. C. (1998) Introduction to Gaussian processes. *NATO ASI Series F (Computer and Systems Sciences)* **168**, 133–166.

Maestrini, L. and Wand, M. P. (2021) The Inverse G-Wishart distribution and variational message passing. *Australian & New Zealand Journal of Statistics* **63**(3), 517–541.

Magnus, J. R. and Neudecker, H. (1988) *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Ltd., Chichester.

Maybeck, P. S. (1979) *Stochastic Models, Estimation, and Control. Vol. I*. Volume 141 of *Mathematics in Science and Engineering*. Academic Press, Inc., New York-London.

Menictas, M., Credico, G. D. and Wand, M. P. (2019) Streamlined variational inference for linear mixed models with crossed random effects. *arXiv preprint arXiv:1910.01799* .

Menictas, M., Nolan, T. H., Simpson, D. G. and Wand, M. P. (2021) Streamlined variational inference for higher level group-specific curve models. *Statistical Modelling* **21**(6), 479–519.

Minka, T. P. (2001a) Expectation Propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 362–369. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Minka, T. P. (2001b) *A family of algorithms for approximate Bayesian inference*. Ph.D. thesis, Massachusetts Institute of Technology.

Minka, T. P. (2004) Power EP. Technical Report MSR-TR-2004-149, Microsoft Research.

Minka, T. P. (2005) Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research.

Minka, T. P. and Winn, J. (2008) Gates: A graphical notation for mixture models. Technical Report MSR-TR-2008-185, Microsoft Research.

Minka, T. P., Winn, J., Guiver, J., Zaykov, Y., Fabian, D. and Bronskill, J. (2018) Infer.NET 0.3. Microsoft Research Cambridge. `http://dotnet.github.io/infer`.

Monahan, J. F. and Stefanski, L. A. (1989) Normal scale mixture approximations to $F^*(z)$ and computation of the logistic-normal integral. In *Handbook of the Logistic Distribution*, pp. 529–540. Marcel Dekker, New York.

Monterrubio-Gómez, K., Roininen, L., Wade, S., Damoulas, T. and Girolami, M. (2020) Posterior inference for sparse hierarchical non-stationary models. *Computational Statistics & Data Analysis* **148**, 106954, 22.

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

Murphy, K. P., Weiss, Y. and Jordan, M. I. (1999) Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pp. 467–475. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Neal, R. M. (2011) MCMC using Hamiltonian dynamics. In *Handbook of Markov chain Monte Carlo*, Chapman & Hall/CRC Handb. Mod. Stat. Methods, pp. 113–162. CRC Press, Boca Raton, FL.

Neal, R. M. and Hinton, G. E. (1998) *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*, pp. 355–368. Dordrecht: Springer Netherlands.

Neville, S. E., Ormerod, J. T. and Wand, M. P. (2014) Mean field variational Bayes for continuous sparse signal shrinkage: pitfalls and remedies. *Electronic Journal of Statistics* **8**(1), 1113–1151.

Nickisch, H. and Rasmussen, C. E. (2008) Approximations for binary Gaussian process classification. *Journal of Machine Learning Research (JMLR)* **9**, 2035–2078.

Nielsen, F. (2020) An elementary introduction to information geometry. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **22**(10), Paper No. 1100, 61.

Nolan, T. H., Menictas, M. and Wand, M. P. (2020) Streamlined computing for variational inference with higher level random effects. *Journal of Machine Learning Research (JMLR)* **21**, Paper No. 157, 62.

Nolan, T. H. and Wand, M. P. (2017) Accurate logistic variational message passing: algebraic and numerical details. *Stat* **6**, 102–112.

Nolan, T. H. and Wand, M. P. (2020) Streamlined solutions to multilevel sparse matrix problems. *The Australian & New Zealand Industrial and Applied Mathematics Journal* **62**(1), 18–41.

Nott, D. J., Tan, S. L., Villani, M. and Kohn, R. (2012) Regression density estimation with variational methods and stochastic approximation. *Journal of Computational and Graphical Statistics* **21**(3), 797–820.

O'Hagan, A. (1978) Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **40**(1), 1–42.

O'Hara, R. B. and Sillanpää, M. J. (2009) A review of Bayesian variable selection methods: what, how and which. *Bayesian Analysis* **4**(1), 85–117.

Ong, V. M.-H., Nott, D. J. and Smith, M. S. (2018) Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics* **27**(3), 465–478.

Opper, M. and Archambeau, C. (2009) The variational Gaussian approximation revisited. *Neural Computation* **21**(3), 786–792.

Opper, M. and Winther, O. (1997) A mean field algorithm for Bayes learning in large feed-forward neural networks. In *Advances in Neural Information Processing Systems*, volume 9. MIT Press.

Ormerod, J. T. and Wand, M. P. (2010) Explaining variational approximations. *The American Statistician* **64**(2), 140–153.

Ormerod, J. T. and Wand, M. P. (2012) Gaussian variational approximate inference for generalized linear mixed models. *Journal of Computational and Graphical Statistics* **21**(1), 2–17.

Ormerod, J. T., You, C. and Müller, S. (2017) A variational Bayes approach to variable selection. *Electronic Journal of Statistics* **11**(2), 3549–3594.

Paciorek, C. J. (2003) *Nonstationary Gaussian Processes for Regression and Spatial Modelling*. Ph.D. thesis. Carnegie Mellon University.

Paciorek, C. J. and Schervish, M. J. (2006) Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics* **17**(5), 483–506.

Paisley, J., Blei, D. M. and Jordan, M. I. (2012) Variational Bayesian inference with stochastic search. ICML'12, pp. 1363–1370. Madison, WI, USA: .

Parisi, G. (1988) *Statistical Field Theory*. Volume 66 of *Frontiers in Physics*. Benjamin/Cummings Publishing Co., Inc., Advanced Book Program, Reading, MA.

Park, S. and Choi, S. (2010) Hierarchical Gaussian process regression. In *Proceedings of 2nd Asian Conference on Machine Learning*, volume 13 of *Proceedings of Machine Learning Research*, pp. 95–110. Tokyo, Japan: PMLR.

Park, T. and Casella, G. (2008) The Bayesian lasso. *Journal of the American Statistical Association* **103**(482), 681–686.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. The Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann, San Mateo, CA.

Peterson, C. and Anderson, J. R. (1987) A mean field theory learning algorithm for neural networks. *Complex Systems* **1**(5), 995–1019.

Pinheiro, J. C. and Bates, D. M. (2006) *Mixed-Effects Models in S and S-Plus*. Springer, New York.

Plagemann, C., Kersting, K. and Burgard, W. (2008) Nonstationary Gaussian process regression using point estimates of local smoothness. In *Machine Learning and Knowledge Discovery in Databases*, pp. 204–219. Berlin, Heidelberg: Springer Berlin Heidelberg.

Plummer, M. *et al.* (2003) JAGS: A program for analysis of Bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, volume 124, pp. 1–10.

Pogány, T. K. and Nadarajah, S. (2021) Explicit forms for three integrals in Wand et al. *Mathematical Communications* **26**(1), 101–105.

Polson, N. G. and Scott, J. G. (2011) Shrink globally, act locally: sparse Bayesian regularization and prediction. In *Bayesian Statistics, 9*, pp. 501–538. Oxford Univ. Press, Oxford.

Quiroz, M., Nott, D. J. and Kohn, R. (2020) Gaussian variational approximation for high-dimensional state space models. *arXiv preprint arXiv:1801.07873* .

R Core Team (2020) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org/`.

Raiko, T., Valpola, H., Harva, M. and Karhunen, J. (2007) Building blocks for variational Bayesian learning of latent variable models. *Journal of Machine Learning Research (JMLR)* **8**, 155–201.

Ranganath, R., Gerrish, S. and Blei, D. (2014) Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 814–822. Reykjavik, Iceland: PMLR.

Rao, J. N. K. and Molina, I. (2015) *Small Area Estimation*. Second edition. Wiley Series in Survey Methodology. John Wiley & Sons, Inc., Hoboken, NJ.

Rasmussen, C. E. and Williams, C. K. I. (2006) *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.

Ray, P. and Bhattacharya, A. (2018) Signal adaptive variable selector for the horseshoe prior. *arXiv preprint arXiv:1810.09004* .

Rényi, A. (1961) On measures of entropy and information. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, pp. 547–561. Univ. California Press, Berkeley, Calif.

Rezende, D. J., Mohamed, S. and Wierstra, D. (2014) Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286. Bejing, China: PMLR.

van Rijsbergen, C. J. (2004) *The Geometry of Information Retrieval*. Cambridge University Press, Cambridge.

Risser, M. D. and Calder, C. A. (2015) Regression-based covariance functions for nonstationary spatial modeling. *Environmetrics* **26**(4), 284–297.

Risser, M. D. and Turek, D. (2020) Bayesian inference for high-dimensional nonstationary Gaussian processes. *Journal of Statistical Computation and Simulation* **90**(16), 2902–2928.

Robbins, H. and Monro, S. (1951) A stochastic approximation method. *Annals of Mathematical Statistics* **22**, 400–407.

Robert, C. P. (1994) *The Bayesian Choice*. Springer Texts in Statistics. Springer-Verlag, New York.

Robert, C. P. and Casella, G. (2004) *Monte Carlo Statistical Methods*. Second edition. Springer Texts in Statistics. Springer-Verlag, New York.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N. and Aigrain, S. (2013) Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London. Series A (Mathematical, Physical and Engineering Sciences)* **371**(1984), 1–25.

Rockafellar, R. T. (1970) *Convex Analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J.

Roeder, G., Wu, Y. and Duvenaud, D. K. (2017) Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Rohde, D. and Wand, M. P. (2016) Semiparametric mean field variational Bayes: general principles and numerical issues. *Journal of Machine Learning Research (JMLR)* **17**, Paper No. 172, 47.

Roininen, L., Girolami, M., Lasanen, S. and Markkanen, M. (2019) Hyperpriors for Matérn fields with applications in Bayesian inversion. *Inverse Problems and Imaging* **13**(1), 1–29.

Roverato, A. (2000) Cholesky decomposition of a hyper inverse Wishart matrix. *Biometrika* **87**(1), 99–112.

Ruppert, D., Wand, M. P. and Carroll, R. J. (2003) *Semiparametric Regression*. Volume 12 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge.

Saha, A., Bharath, K. and Kurtek, S. (2020) A geometric variational approach to Bayesian inference. *Journal of the American Statistical Association* **115**(530), 822–835.

Salimans, T. and Knowles, D. A. (2013) Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis* **8**(4), 837–881.

Sampson, P. D. and Guttorp, P. (1992) Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association* **87**(417), 108–119.

Sanderson, C. and Curtin, R. (2016) Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software* **1**(2).

Schelldorfer, J., Bühlmann, P. and van de Geer, S. (2011) Estimation for high-dimensional linear mixed-effects models using ℓ_1-penalization. *Scandinavian Journal of Statistics. Theory and Applications* **38**(2), 197–214.

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

Seeger, M. (2000) Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Sikorska, K., Rivadeneira, F., Groenen, P. J., Hofman, A., Uitterlinden, A. G., Eilers, P. H. and Lesaffre, E. (2013) Fast linear mixed model computations for genome-wide association studies with longitudinal data. *Statistics in Medicine* **32**(1), 165–180.

Smola, A. and Bartlett, P. (2001) Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Smyth, G., Hu, Y., Dunn, P., Phipson, B. and Chen, Y. (2021) *statmod: Statistical Modeling*. R package version 1.4.36. `https://CRAN.R-project.org/package=statmod`.

Snow, G. (2020) *TeachingDemos: Demonstrations for Teaching and Learning*. R package version 2.12. `https://CRAN.R-project.org/package=TeachingDemos`.

Stan Development Team (2020) *RStan: the R interface to Stan*. R package version 2.21.1. `http://mc-stan.org/`.

Stegle, O., Fallert, S. V., MacKay, D. J. C. and Brage, S. (2008) Gaussian process robust regression for noisy heart rate data. *IEEE Transactions on Biomedical Engineering* **55**(9), 2143–2151.

Stein, M. L. (1999) *Interpolation of Spatial Data*. Springer Series in Statistics. Springer-Verlag, New York.

Tan, L. S. L., Bhaskaran, A. and Nott, D. J. (2020) Conditionally structured variational Gaussian approximation with importance weights. *Statistics and Computing* **30**(5), 1255–1272.

Tan, L. S. L. and Nott, D. J. (2018) Gaussian variational approximation with sparse precision matrices. *Statistics and Computing* **28**(2), 259–275.

Tang, X., Ghosh, M., Xu, X. and Ghosh, P. (2018) Bayesian variable selection and estimation based on global-local shrinkage priors. *Sankhya A. The Indian Journal of Statistics* **80**(2), 215–246.

Taylor, P. M. (1980) The First Year of Life: The Collaborative Perinatal Project of the National Institute of Neurological and Communicative Disorders and Stroke. *Journal of the American Medical Association* **244**(13), 1503–1503.

Thompson, P. D. (1956) Optimum smoothing of two-dimensional fields. *Tellus* **8**, 384–393.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **58**(1), 267–288.

Titsias, M. and Lawrence, N. D. (2010) Bayesian Gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 844–851. Chia Laguna Resort, Sardinia, Italy: PMLR.

Titsias, M. and Lázaro-Gredilla, M. (2014) Doubly stochastic variational Bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1971–1979. Bejing, China: PMLR.

Titterington, D. M. (2004) Bayesian methods for neural networks and related models. *Statistical Science* **19**(1), 128–139.

Tolvanen, V., Jylänki, P. and Vehtari, A. (2014) Expectation propagation for nonstationary heteroscedastic Gaussian process regression. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6.

Tung, D. T., Tran, M.-N. and Cuong, T. M. (2019) Bayesian adaptive lasso with variational Bayes for variable selection in high-dimensional generalized linear mixed models. *Communications in Statistics. Simulation and Computation* **48**(2), 530–543.

Uhler, C., Lenkoski, A. and Richards, D. (2018) Exact formulas for the normalizing constants of Wishart distributions for graphical models. *The Annals of Statistics* **46**(1), 90–118.

van der Vaart, A. W. and van Zanten, J. H. (2008) Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics* **36**(3), 1435–1463.

de Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Temple Lang, D. and Bodik, R. (2017) Programming with models: writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics* **26**(2), 403–413.

Vehtari, A., Gelman, A., Sivula, T., Jylänki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D. and Robert, C. P. (2020) Expectation propagation as a way of life: a framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research (JMLR)* **21**, Paper No. 17, 53.

Verbeke, G. and Molenberghs, G. (2000) *Linear Mixed Models for Longitudinal Data*. Springer Series in Statistics. Springer-Verlag, New York.

Vonesh, E. F. and Chinchilli, V. M. (1997) *Linear and Nonlinear Models for the Analysis of Repeated Measurements*. Volume 154 of *Statistics: Textbooks and Monographs*. Marcel Dekker, Inc., New York.

Wainwright, M. J. and Jordan, M. I. (2008) Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* **1**(1–2), 1–305.

Wand, M. P. (2002) Vector differential calculus in statistics. *The American Statistician* **56**(1), 55–62.

Wand, M. P. (2014) Fully simplified multivariate normal updates in non-conjugate variational message passing. *Journal of Machine Learning Research (JMLR)* **15**, 1351–1369.

Wand, M. P. (2017) Fast approximate inference for arbitrarily large semiparametric regression models via message passing. *Journal of the American Statistical Association* **112**(517), 137–156.

Wand, M. P. (2020) *KernSmooth: Functions for Kernel Smoothing Supporting Wand and Jones (1995)*. R package version 2.23-18. `https://CRAN.R-project.org/package=KernSmooth`.

Wand, M. P. and Jones, M. C. (1995) *Kernel Smoothing*. Volume 60 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, Ltd., London.

Wand, M. P., Ormerod, J. T., Padoan, S. A. and Frührwirth, R. (2011) Mean field variational Bayes for elaborate distributions. *Bayesian Analysis* **6**(4), 847–900.

Wang, B. and Titterington, D. M. (2006) Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model. *Bayesian Analysis* **1**(3), 625–649.

Wang, S. S. J. and Wand, M. P. (2011) Statistical computing and graphics using Infer.NET for statistical analyses. *The American Statistician* **65**(2), 115–126.

Wang, Y. and Blei, D. M. (2019) Frequentist consistency of variational Bayes. *Journal of the American Statistical Association* **114**(527), 1147–1161.

Warnes, J. J. and Ripley, B. D. (1987) Problems with likelihood estimation of covariance functions of spatial Gaussian processes. *Biometrika* **74**(3), 640–642.

West, M. (1987) On scale mixtures of normal distributions. *Biometrika* **74**(3), 646–648.

Whittaker, E. T. and Watson, G. N. (1996) *A Course of Modern Analysis*. Cambridge Mathematical Library. Cambridge University Press, Cambridge.

Wilkinson, W., Chang, P., Andersen, M. and Solin, A. (2020) State space expectation propagation: Efficient inference schemes for temporal Gaussian processes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10270–10281. PMLR.

Wingate, D. and Weber, T. (2013) Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299* .

Winn, J. and Bishop, C. M. (2005) Variational message passing. *Journal of Machine Learning Research (JMLR)* **6**, 661–694.

Xu, M., Quiroz, M., Kohn, R. and Sisson, S. A. (2019) Variance reduction properties of the reparameterization trick. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2711–2720. PMLR.

Yang, M. (2013) Bayesian nonparametric centered random effects models with variable selection. *Biometrical Journal* **55**(2), 217–230.

Yang, M., Wang, M. and Dong, G. (2020) Bayesian variable selection for mixed effects model with shrinkage prior. *Computational Statistics* **35**(1), 227–243.

You, C., Ormerod, J. T. and Müller, S. (2014) On variational Bayes estimation and variational information criteria for linear regression models. *Australian & New Zealand Journal of Statistics* **56**(1), 73–87.

Zeiler, M. D. (2012) ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Zhang, C., Butepage, J., Kjellstrom, H. and Mandt, S. (2019) Advances in variational inference. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **41**(8), 2008–2026.

Zhang, F. and Gao, C. (2020) Convergence rates of variational posterior distributions. *The Annals of Statistics* **48**(4), 2180–2207.

Zhang, Y. and Bondell, H. D. (2018) Variable selection via penalized credible regions with Dirichlet-Laplace global-local shrinkage priors. *Bayesian Analysis* **13**(3), 823–844.

Zhao, Y., Staudenmayer, J., Coull, B. A. and Wand, M. P. (2006) General design Bayesian generalized linear mixed models. *Statistical Science* **21**(1), 35–51.

Zhao, Z., Emzir, M. and Särkkä, S. (2021) Deep state-space Gaussian processes. *Statistics and Computing* **31**(6), 75.

Zhu, H. and Rohwer, R. (1997) *Measurements of Generalisation Based on Information Geometry*, pp. 394–398. Boston, MA: Springer US.

Zou, H. (2006) The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101**(476), 1418–1429.

Zou, H. and Hastie, T. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67**(2), 301–320.

Emanuele Degani

CURRICULUM VITAE

Personal Details

Date and Place of Birth: August 9th, 1994 – Brescia (Italy)
Nationality: Italian

Contact Information

University of Padova, Department of Statistics
via Cesare Battisti, 241-243
35121 Padova. Italy.

Tel. +39 049 827 4141
e-mail: emanuele.degani@phd.unipd.it; emanuele.achab@gmail.com (personal)

Current Position

Since October 2018 (expected completion: January 2022)
PhD Student in Statistical Sciences, University of Padova.
Thesis title: *Some developments on variational approximation methods for Bayesian inference.*
Supervisor: Prof. Mauro Bernardi
Co-supervisor: Luca Maestrini

Research interests

- Variational approximations
- Bayesian inference
- Computational statistics
- Machine learning

Education

October 2016 – September 2018
Master degree (*laurea magistrale***) in Statistical Sciences**.
University of Padova, Faculty of Statistical Sciences
Title of dissertation: "Bayesian inference for Support Vector Machines mixture models"
Supervisor: Prof. Mauro Bernardi
Final mark: 110/110 cum Laude

October 2013 – April 2016
Bachelor degree (*laurea triennale***) in Statistics, Economics and Finance**.
University of Padova, Faculty of Statistical Sciences
Title of dissertation: "Monoplex to Multiplex networks analysis generalization: formalization, description and implementation of the commonest measures via a statistical package"
Supervisor: Prof. Giovanna Menardi
Final mark: 110/110 cum Laude

September 2008 – July 2013
High school diploma (*diploma d'istruzione superiore***)**.
Istituto Tecnico Commerciale Abba-Ballini, Brescia, Italy
Final mark: 100/100

Work experience

September 2016 – April 2017
Undegraduate research fellow.
Ca' Foscari University of Venezia, Department of Economics. Under the supervision of Prof. Stefano Campostrini.

Teaching experience

November 2020 – January 2021
Applied Statistics
Master Degree in Molecular Biology
Teaching assistance on R classes, 21 hours
Department of Biology, University of Padova
Instructor: Prof. Alessandra Rosalba Brazzale

November 2019 – January 2020
Applied Statistics
Master Degree in Molecular Biology
Teaching assistance on R classes, 21 hours
Department of Biology, University of Padova
Instructor: Prof. Alessandra Rosalba Brazzale

September 2017 – September 2018
Calculus 1 and Advanced Statistics
Academic tutor
Department of Statistical Sciences, University of Padova

Awards and Scholarship

2016
Mille e una lode. Study grant reserved to the 3% *most valuable* students for each degree course.

2014
Academic yearly tuition fees refund for first-year students with GPA higher than 28/30.

Computer skills

- Good knowledge of R
- Good knowledge of LaTeX
- Good knowledge of Microsoft Office
- Basic knowledge of C++ and related R libraries
- Basic knowledge of Python
- Basic knowledge of HTML

Language skills

Italian: native; English: fluent (written/spoken); Spanish: basic (written/spoken).

Publications

Articles in conference proceedings
Degani, E., Maestrini, L. and Bernardi, M. (2021). Model fitting in Bayesian inference via Power Expectation Propagation. In *Book of Short Paper SIS 2021*, (Perna, C., Salvati, N. and Schirripa Spagnolo, F.) pp. 1026–1031, Pearson, ISBN: 9788891927361.

Working papers
Degani, E., Maestrini, L., Toczydłowska, D. and Wand, M. P. (2021). Sparse linear mixed model selection via streamlined variational Bayes. Unpublished manuscript publicly available at `https://arxiv.org/abs/2110.07048`.

Conference presentations

Degani, E., Maestrini, L., Toczydłowska, D. and Wand, M. P. (2021). Sparse linear mixed model selection via streamlined variational Bayes. *14th International Conference of the ERCIM W.G. on Computational and Methodological Statistics*, London, U.K., 18–20 December. Organized invited session #B0839.

Degani, E., Maestrini, L. and Bernardi, M. (2021). Model fitting in Bayesian inference via Power Expectation Propagation (oral presentation). *50th Scientific Meeting of the Italian Statistical Society*, Pisa, Italy, 21–25 June.

Degani, E. (2020). Monitorare l'evoluzione della pandemia in Italia in tempo reale (oral presentation, publicly available at `https://youtu.be/dmXGMLgPLmI`). *VenetoNight Padova*, Padova, Italy, 27 December.

Degani, E. and Busatto, C. (2018). Deep Learning models for the *top-pair* production process identification (poster and three minutes oral presentation). *Workshop on Advanced Statistics for Physics Discovery*, Padova, Italy, 24–25 September.

References

Prof. Mauro Bernardi
University of Padova, Department of Statistical Sciences
Address: Via Cesare Battisti 241, Padova, Italy
Phone: +39 049 8274165
e-mail: mauro.bernardi@unipd.it

Luca Maestrini
Australian National University, Research School of Finance, Actuarial Studies and Statistics
Address: Building 26C, Kingsley Street, Canberra ACT 2601, Australia
e-mail: luca.maestrini@anu.edu.au