Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for naturvitenskap
Institutt for matematiske fag

Vegard Kolaas

Penalised complexity priors in hierarchical models

Masteroppgave i Applied Physics and Mathematics
Veileder: Geir-Arne Fuglstad
Januar 2022

NTNU
Kunnskap for en bedre verden

Vegard Kolaas

Penalised complexity priors in hierarchical models

Masteroppgave i Applied Physics and Mathematics
Veileder: Geir-Arne Fuglstad
Januar 2022

Norges teknisk-naturvitenskapelige universitet
Fakultet for naturvitenskap
Institutt for matematiske fag

NTNU
Kunnskap for en bedre verden

Abstract

Hierarchical decomposition (HD) priors are a prior construction framework for setting joint priors on variance parameters in latent Gaussian models (LGMs) using a tree-structure reflecting the structure of the model. Currently they can only incorporate random effects, not fixed effects, meaning they can at most decompose the residual variance after linear regression.

In this thesis we aim to remove this limitation, extending the framework to also include fixed effects, and testing these new priors on a series of non-linear smoothing problems, in 1 and 2 dimensions, with complete and sparse data sets. In all cases we evaluate prior performance using continuous rank probability scores and mean square errors.

Our findings are inconclusive regarding whether HD priors, with or without incorporating fixed effect variance, are an improvement over competing priors performance-wise. Whether HD, expanded HD or independent priors perform the best varies between tests. Further research is needed to reach a general conclusion.

Applying the new priors to multidimensional smoothing problems can cause impractical runtimes compared to independent or basic HD priors, though it is possible this is more due to the complexity of the HD tree than the framework expansion per se. We therefore recommend further research into performing inference with the expanded HD priors before using them in practice.

Contents

Chapter 1

Introduction

Bayesian statistics is a highly flexible tool that, owing to powerful statistical software like WinBUGS(Spiegelhalter et al., 2000) and Stan(Carpenter et al., 2017) sees a wide range of applications across all statistical fields of science, and within Bayesian statistics, latent Gaussian models (LGMs) are used particularly frequently. The defining property of this class of models is that all latent parameters are normally distributed. At first it might sound impractically limiting to restrict oneself to a single model class, but, as users of INLA (Rue et al., 2009) can attest (Rue et al., 2016), this is nowhere near the issue one might first assume.

However, the field of applying these models is still far from being solved, with prior choice potentially being a particular challenge. Fuglstad et al. (2020) attempts to mitigate part of this problem by offering an intuitive framework for setting joint priors on variance parameters in LGMs, namely hierarchical decomposition (HD) priors, which we will be expanding upon in this thesis. These priors hold much promise. They are intuitive to specify and reflect model structure in a transparent way. However, Fuglstad et al. (2020) noted one remaining step in particular; the priors can currently only handle variance from random effects, not fixed ones. We will expand the framework so that they can, provide new theory and explore how viable the expanded framework is through a series of examples.

For all of our examples, priors are compared by applying them to non-linear smoothing problems. We begin by comparing basic HD priors to competing priors on a 1-dimensional problem with a rich data set, with ideal as well as poor prior beliefs. We then introduce the expanded HD prior, that incorporates fixed effect variance, offering theory on how to conceptualise model and node variance in this new context, before we compare them to the other priors on the same one dimensional problem, first with a rich data set, then two different sparse ones.

Finally, we perform a simulation study on a 2-dimensional problem with sparse data sets. In all cases, objective prior performance is measured using MSE and CRPS scores. We will also consider the aspect of how intuitive the different priors are to specify.

HD priors are part of an on-going effort to supply general practitioners with default priors. In their paper on penalised complexity (PC) priors, Simpson et al. (2017) noted that, as developers of INLA, they were faced with an unpalatable choice between requiring that users specify full joint priors for model parameters on their own, or to supply users with default priors. Neither alternative was particularly inviting. Despite being the mathematically correct option, leaving prior choice entirely up to end users would not be feasible due to all the confusion that would ensue. Default priors meanwhile were problematic in that they were often chosen somewhat arbitrarily in hopes that they would provide decent results. PC priors can then, while not universal, be seen as an attempt to mitigate this problem by providing a general means to specify priors that are both conservative and intuitive to use, as well as having a number of other desirable properties. In light of this HD priors are an extension of the PC prior framework to the more narrow domain of joint variance priors for LGMs.

One may also draw parallels between HD priors and the R2-D2 shrinkage prior (Yanchenko et al., 2021), indeed Yanchenko et al. (2021) does this themselves in their introduction, conceptualising Simpson et al. (2017)'s multivariate PC priors as a means of performing shrinkage on the entire model. They also employ a similar scheme of variance decomposition, assigning a Dirichlet prior on portions of variance corresponding to each random effect model component. This is similar to assigning an unstructured prior in the HD framework. However, the resulting prior seems more ad hoc, which runs counter to calls for Bayesian workflow (Gelman et al., 2020) in which model understanding is prioritised. To this end, HD priors seem more appropriate, as they respect model structure and makes inputting prior knowledge more intuitive.

In chapter 2 we provide necessary background material, laying out our general model, the principled priors we will be expanding upon and the algorithm and software we will be make use of in performing inference with said model. In Chapter 3, we introduce our main example problem along with our basic priors. Chapter 4 is where we introduce the expanded HD prior. Here we also provide additional theory regarding how to conceptualise total (node) variance in this new context. Chapter 5 is our first step towards more realistic model complexity, with priors being tested on sparse data sets, and Chapter 6 takes this a step further, with a simulation study on applying the different priors to sparse data sets for a 2-dimensional problem.

Chapter 2

Background

2.1 Bayesian hierarchical models

Central to Bayesian inference are (Bayesian) hierarchical models, for a number of reasons. Firstly, realistic complexity often entails a hierarchy of variables. In many cases, they are also simply more representative of the truth. The archetypical example is standardised school test scores, which have multiple levels of experimental units, from individual students, to classes, to schools, to districts, to countries. Even without a strictly defined formal hierarchy, inference can still be helped by finding ways of grouping observations to utilise more information. On a more human level, hierarchical models may also simply be more intuitive, with relationships between model parameters that are more easily understood, in line with recommended Bayesian workflow(Rue et al., 2016). One specific example, and the main model we will using for this thesis is the non-linear smoothing model.

Example 1. The general non-linear smoothing model: Let $\boldsymbol{y}$ be a signal with normally distributed random noise, and linear and non-linear contributions from p covariates $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_p\}$. Assuming that any non-linear contribution is continuously differentiable, and that there is no interaction between covariates, we can model $\boldsymbol{y}$ as follows:

$$\boldsymbol{y} \mid \sigma_R, \alpha, \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\eta}, \sigma_R^2) \tag{2.1}$$

$$\boldsymbol{\eta} = \alpha \mathbf{1} + \sum_{i=1}^{p} f_i(\boldsymbol{x}_i) \tag{2.2}$$

$$\boldsymbol{f}_i(\boldsymbol{x}_i) = \beta_i \boldsymbol{x}_i + \boldsymbol{u_i} \tag{2.3}$$

$$\alpha \sim \mathcal{N}(0, \sigma_\alpha^2) \tag{2.4}$$

$$\beta_i \sim \mathcal{N}(0, \sigma_{\beta_i}^2)\,,\ i = 1, 2, ..., p \tag{2.5}$$

$$\boldsymbol{u}_i \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{u}_i}^2 \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_i})\,,\ i = 1, 2, ..., p \tag{2.6}$$

with additional priors on the set of variance parameters $\{\sigma_\alpha, \sigma_{\beta_1}, \sigma_{\beta_2}, ..., \sigma_{\beta_p}, \sigma_{\boldsymbol{u}_1}, \sigma_{\boldsymbol{u}_2}, ..., \sigma_{\boldsymbol{u}_p}, \sigma_R\}$. Here α is the population intercept, $\boldsymbol{\beta}$ is the vector of fixed effects, and the vectors $\boldsymbol{u}_i\,,\ i = 1, 2, ..., p$ are random-walk-2 vectors representing the non-linear contributions from each covariate. The distribution of $\boldsymbol{u}$ is explained more closely in Section 2.2.1, and our variance priors in general are discussed in Section 2.3.2.

Although this model has a total of p covariates, it is still substantially limited in the range of behaviour it can capture as it assumes no interaction between covariates. Nevertheless it is sufficiently complex to be used in interesting problems within the scope of this thesis, and will indeed generally be the model by which data is generated and modelled, with the exception of Chapter 3, in which we explicitly encode a specific non-linear function when generating data.

This layered approach is highly flexible, easing the process of encoding desired knowledge into the model, and overall the approach is versatile, as the hierarchy can be adjusted depending on the specific problem. Although models set up this way are generally not analytically tractable, this problem is typically mitigated by the availablity of powerful Bayesian software for approximating the posterior using Markov chain Monte Carlo methods. The main software we will be using is Stan, which, as we shall discuss further in Section 2.4, involves a particularly sophisticated approach to MCMC sampling, namely the No-U-turn sampler.

2.2 Gaussian Markov random fields

Out of the model components mentioned so far, the most complex, and thus the potentially most expensive to simulate as part of our inference, are the random vectors used to represent non-linear effects mentioned in Example 1. Performing Markov chain Monte Carlo-based Bayesian inference using a multivariate normal distribution requires computing the inverse of that distribution's covariance matrix, its precision matrix. In general, inverting matrices can quickly get very

expensive, so mitigating this cost is an important part of keeping this type of inference, where the posterior is evaluated thousands of times, practically feasible. One way to achieve this for LGMs, is to ensure the precision matrices are sparse. As we shall demonstrate in this section, this holds for our random vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_p$, as they belong to the category of Gaussian Markov random fields (GMRFs).

Gaussian Markov random fields encompasses a wide array of statistical models, with applications in spatiotemporal statistics, analysis of time-series and longitudinal and survival data, as well as semi-parametric statistics and graphical data(Rue and Held, 2005), but they are not least particularly relevant to Bayesian inference using hierarchical models. They provide naturally sparse precision matrices (Rue and Held, 2005), which is extremely useful for MCMC procedures on LGMs due to the aforementioned costs that might otherwise arise.

The key property that distinguishes GMRFs from Gaussian random vectors in general is that their full conditional dependence structure can be described with an undirected graph. More specifically:

Definition 2.2.1. Gaussian Markov random field: If $\boldsymbol{u} \in \mathbf{R}^n$, and $\mathcal{G}$ is an undirected graph consisting of a set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of edges, then $\boldsymbol{u}$ is a GMRF with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{Q}$ if and only if it has density

$$\pi(\boldsymbol{u}) = (2\pi)^{-\frac{n}{2}} \mid \boldsymbol{Q} \mid^{1/2} \exp(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu})^T \boldsymbol{Q}(\boldsymbol{u}-\boldsymbol{\mu})) \, \forall \boldsymbol{u} \in \mathbf{R}$$

where $\mid \boldsymbol{Q} \mid$ is the determinant of $\boldsymbol{Q}$ and $\boldsymbol{Q}_{ij} \neq 0 \iff \{i,j\} \in \mathcal{E} \, \forall \, i \neq j$.

Given this definition, a number of useful properties follow, besides those that apply to any normal distribution. For instance it can be shown that the conditional dependence structure of $\boldsymbol{x}$ can easily be found from inspecting $\boldsymbol{Q}$.

Theorem 1. *Let $\boldsymbol{u}$ be a GMRF. Then $u_i \perp x_j \iff \boldsymbol{Q}_{ij} = 0$ for $i \neq j$.*

For details, see Rue and Held (2005). This result is a prime example of why GMRFs are desirable, as it means that a sparse dependence structure between covariates also means that evaluating the associated prior is relatively inexpensive.

2.2.1 Intrinsic Gaussian Markov random fields (IGMRFs)

We noted at the start of this section that we are interested in GMRFs for the purpose of modelling the random vectors, $\boldsymbol{u}_i$, $i = 1, 2, ..., p$ that we use for nonlinear smoothing. These vectors actually fall within a specific subclass of GMRFs, namely GMRFs that are *intrinsic*. These GMRFs are characterised by precision matrices without full rank. The definition of such a GMRF is quite similar to that of GMRFs in general:

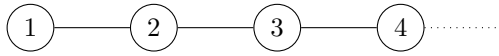


Figure 2.1: The beginning of the graph corresponding to a random-walk 1 model.

Definition 2.2.2. Intrinsic Gaussian Markov random field(IGMRF): Let $\boldsymbol{Q}$ be an symmetric, semi-positive definite $n \times n$ matrix with rank $n-k$, then $\boldsymbol{u}$ is an IGMRF with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{Q}$ if it has density

$$\pi(\boldsymbol{u}) = (2\pi)^{-\frac{n-k}{2}} (\mid \boldsymbol{Q} \mid^*)^{1/2} \exp(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu})^T \boldsymbol{Q}(\boldsymbol{u}-\boldsymbol{\mu}))$$

where $\mid \boldsymbol{Q} \mid^*$ is the generalised determinant of $\boldsymbol{Q}$, ie. the product of all the non-zero eigenvalues of $\boldsymbol{Q}$. Furthermore, $\boldsymbol{u}$ is an IGMRF with respect to the graph $\mathcal{G}$ if and only if $\boldsymbol{Q}_{ij} \neq 0 \iff \{i,j\} \in \mathcal{E} \, \forall \, i \neq j$ We also define the *order* of $\boldsymbol{u}$ to be k.

Note that $\boldsymbol{\mu}$ and $\boldsymbol{Q}$ are not truly the mean and precision matrix of $\boldsymbol{u}$ as they formally do not exist, but we will continue to refer to them as such for convenience when describing IGMRFs. IGMRFs of order k are often constructed using forward differences of order k.

Definition 2.2.3. Forward difference

Let $f(z)$ be a function defined over a regular grid with step length h. Then the forward difference of $f(z)$, $\Delta f(z)$, is given by the following.

$$\Delta f(z) = f(z+h) - f(z)$$

Furthermore, forward differences of higher order are defined recursively,

$$\Delta^k f(z) = \Delta \Delta^{k-1} f(z)$$

so we have for instance $\Delta^2 f(z) = f(z+2h) - 2f(z+h) + f(z)$ and, in the general case $\Delta^k f(z) = (-1)^k \sum_{j=0}^{k} \binom{k}{j} f(z+jh)$.

Example 1. Random-walk-1 model: If we define $\boldsymbol{u} \in \mathbf{R}^N$ and $\Delta u_i \sim \mathcal{N}(0, \sigma)$ for $i = 1, 2, ..., N-1$, then we have a random walk model of order 1. Note that we only define the distribution of the first $N-1$ differences as, for a random walk of order k, the k-order forward difference is not defined for the last k components of $\boldsymbol{u}$, and $k = 1$ in this case. The conditional dependence graph is shown in Figure 2.1.

The precision matrix is straightforward to derive. We need only note that the joint probability density of all the finite differences and the density of $\boldsymbol{u}$ must

be equivalent. Because the normalisation constants are the same, the only parts that differ are the exponents, and, having set up that equality, it is simple to derive $\boldsymbol{Q}$.

In other words, $\pi(\Delta^k u_i, i = 1, 2, ..., n-1) \propto \exp(-\frac{1}{2\sigma}\sum_{i=1}^{n-1}(\Delta^k u_i)^2)$ and $\pi(\boldsymbol{u}) \propto \exp(-\frac{1}{2}\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u})$. Define the structure matrix $\boldsymbol{R} = \boldsymbol{Q}/\sigma$, and the $\boldsymbol{Q}$ follows thus.

$$\sum_{i=1}^{n-1}(\Delta u_i)^2 = (\boldsymbol{D}\boldsymbol{u})^T(\boldsymbol{D}\boldsymbol{u}) = \boldsymbol{u}^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{u} = \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} = \sigma\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u}$$

The resulting precision matrix is then as follows.

$$\boldsymbol{Q} = \frac{1}{\sigma}\begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & -1 & 2 & 1 & \\ & & -1 & 1 & \end{bmatrix}$$

This simple example illustrates the basic ideas of order n random walk models, but is uninteresting for our modelling purposes, as it essentially describes a discretised Brownian motion. However, we do not need much more complexity for the resulting IGMRF to be useful, as the type we use in our model is only one order higher.

Example 2. Random-walk-2 model: The conditional dependence structure of this matrix is as shown in Figure 2.2, and, by a derivation analogous to the one in Example 1, the precision matrix is the following.

$$\boldsymbol{Q} = \begin{bmatrix} 1 & -2 & 1 & & & & & \\ -2 & 5 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & 1 & -4 & 6 & -4 & 1 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 5 & -2 \\ & & & & & 1 & -2 & 1 \end{bmatrix}$$

In either of these cases, the ensuing precision matrices are rank deficient, but can be made proper by introducing constraints. In general, if $\boldsymbol{u} \in \mathbf{R}^n$ is an IGMRF with precision matrix $\boldsymbol{Q}$ with diagonal decomposition $\boldsymbol{Q} = \Sigma_i\lambda_i\boldsymbol{e}_i\boldsymbol{e}_i^T = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ where λ_i is the ith non-zero eigenvalue of $\boldsymbol{Q}$ and $\boldsymbol{e}_i$ is the corresponding

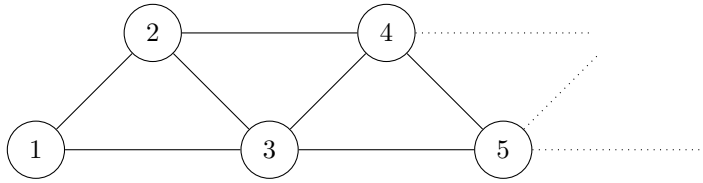


Figure 2.2: The beginning of the graph corresponding to a random walk model of order 2.

eigenvector, then, under the linear constraint $\boldsymbol{Au} = \boldsymbol{a}$ where $\boldsymbol{A}^T = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_k]$ (the specific form of A is not actually a limitation), then we have that

$$\ln \pi(\boldsymbol{u} \mid \boldsymbol{Au} = \boldsymbol{a}) = \frac{n-k}{2} \ln 2\pi + \frac{1}{2} \sum_{i=k+1}^{n} \ln \lambda_i - \frac{1}{2} \boldsymbol{u}^T \tilde{\boldsymbol{Q}} \boldsymbol{u}$$

where $\tilde{\boldsymbol{Q}} = \boldsymbol{V} \tilde{\boldsymbol{\Lambda}} \boldsymbol{V}^T$ where $\tilde{\boldsymbol{\Lambda}} = diag(0, ..., 0, \lambda_{k+1}, ..., \lambda_n)$. For details, see Rue and Held (2005).

2.3 Principled priors

Here, we will, as briefly mentioned in Example 1 in Section 2.1, discuss our variance priors. These priors, HD priors, will be the main focus of this thesis. HD priors are an extension of a more general prior framework, penalised complexity (PC) priors. Both frameworks are derived from a set of principles chosen to elicit desirable prior properties, hence the name of this section.

2.3.1 Penalised complexity priors

We will begin our discussion with PC priors. PC priors were first introduced by Simpson et al. (2017). They noted a need for default priors, and PC priors, though not universal, were an attempt to bridge part of that gap. Simpson et al. (2017) found that these priors were often a step in the right direction. The PC prior framework aims, as the name suggests, to avoid models that are more complex than what is needed to explain the data by denoting a base model and setting priors that penalise deviations from this model.

To illustrate some of the basic principles behind PC priors, we will briefly go through the process of setting a prior on the standard deviation, σ, on a single Gaussian effect. The first principle is that of the aforementioned *base model*, which is typically the simplest model, or at least the the one towards which we wish to enforce shrinkage. For the single Gaussian effect, the base model is

simply the limiting case where $\sigma \to 0$. In order to enforce shrinkage towards the base model, Simpson et al. (2017) defines a "distance" function, which is defined using the Kullback-Leibler divergence (KLD) between the base model, g, and more flexible model f for which σ is non-zero.

$$\text{KLD}(f||g) = \int f(\boldsymbol{x}) \ln \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} = \text{E}_f \left[\ln \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \right]$$

This is in other words the expectation of $\ln \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}$ when $\boldsymbol{x} \sim f$. The expression can be seen as a measure of the information lost when using the simpler distribution g to approximate the more flexible, and thus more complex, f. In general, given the KLD between f and g, the distance function d of the *complexity parameter*, ξ(σ in our specific example), is defined as follows

$$d(\xi) = \sqrt{2\text{KLD}(f||g)}$$

where the 2-factor has been added for convenience, and the square root compensates for the power of two usually associated with KLD. For our example this also yields the intuitive result that $d = \sigma$. For more details, see Simpson et al. (2017).

Note that this function is not formally a distance, as it is not a metric. Nevertheless we will, for simplicity, adopt the terminology used by Simpson et al. (2017) and continue to refer to it as such. Given the distance function, the remaining steps follow easily via the prior on d, $\pi(d)$. Recall that the general aim of PC priors is to penalise complexity, identified as distance from the base model. Therefore, $\pi(d)$ must have a maximum at $d = 0$, and decay with increasing d. Furthermore, because users will generally not be expected to have domain-specific knowledge of $\pi(d)$, the rate of decay is chosen to be constant. This entails a exponential distribution on d, possibly truncated in general depending on the model, and the prior on ξ follows from substitution on $\pi(d)$. For $\xi = \sigma$, this is trivial, but we could also have chosen to set a prior on variance, σ^2, precision, σ^{-2}, or any other bijective transformation of σ and obtained an equivalent result, as they all would have followed from $\pi(d)$. This is a nother noteworthy property of PC priors, *invariance to reparameterisation.*

2.3.2 Hierarchical variance decomposition

Originally formulated by Fuglstad et al. (2020), the HD prior framework uses, in part, PC priors to decompose the total latent variance in latent Gaussian models. This framework will be the main focus on this thesis, as we test some basic HD priors in Chapter 3, expand upon it by accounting for fixed effect variance in Chapter 4 and test these new priors in chapters 5 and 6.

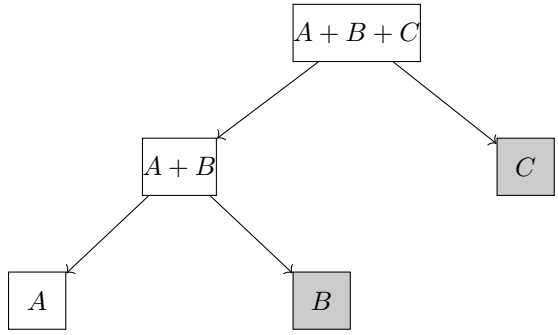


Figure 2.3: An example of a typical HD hierarchy structure. Grey colouration denotes preferred nodes.

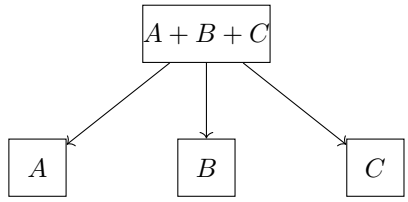


Figure 2.4: An example of an unstructured HD hierarchy. Generally used to express ignorance.

To summarise, given a model with a set of associated variance parameters, Fuglstad et al. (2020) assigns a prior to the total model variance, and parameterises the variances corresponding to specific model components as proportions thereof, which he assigns using a series of splits. The eponymous variance decomposition hierarchy is then given by the ordering and types of these splits. These hierarchies can be visualised as (binary) trees, see for instance Figures 2.3 and 2.4.

More specifically, the total model variance, V, is given a typical variance prior, for instance a PC prior like the one described in Section 2.3.1. The first split in the hierarchy is then between the total latent model variance, $t = \sum_{i=1}^{N} \sigma_i^2$ and the residual variance σ_R^2, and is controlled by the parameter $\omega_R = \frac{\sigma_R^2}{t+\sigma_R^2}$ and its assigned prior. Given this top split, the rest of the hierarchy is up to the specific researcher. Although it is redundant to assign a parameter ω_i for every σ_i^2 , $i = 1, 2, ..., N$, we will still denote portions of variance as if we do for simplicity's sake.

Splits can either express ignorance, or some degree of information. In the

latter case Fuglstad et al. (2020) suggests using binary splits with PC priors, which are given in the following theorem.

Theorem 2. *Let $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ be random effects entering into the linear predictor through $\boldsymbol{A}_i\boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, \sigma_i^2\tilde{\boldsymbol{\Sigma}}_i)$, $i = 1, 2$*

Then if $\tilde{\boldsymbol{\Sigma}}_1 + \tilde{\boldsymbol{\Sigma}}_2$ is invertible then $\omega = \sigma_2^2/(\sigma_1^2 + \sigma_2^2)$ has the following priors. If the base model is $\omega_0 = 0$ then

$$\pi(\omega) = \begin{cases} \frac{\lambda|d'(\omega)|}{1-\exp(-\lambda d(1))}\exp(-\lambda d(\omega)) & , \tilde{\boldsymbol{\Sigma}}_1 \textit{invertible} \\ \frac{\lambda}{2\sqrt{\omega}(1-\exp(-\lambda))}\exp(-\lambda\sqrt{\omega}) & , \tilde{\boldsymbol{\Sigma}}_1 \textit{singular} \end{cases}, \omega \in (0, 1)$$

If ω_0 is set to a median value, ω_m, then

$$\pi(\omega) = \begin{cases} \frac{\lambda|d'(\omega)|}{2[1-\exp(-\lambda d(0))]}\exp(-\lambda d(\omega)) & , \omega \in (0, \omega_0) \\ \frac{\lambda|d'(\omega)|}{2[1-\exp(-\lambda d(1))]}\exp(-\lambda d(\omega)) & , \omega \in (\omega_0, 1) \end{cases}$$

In both of these cases the distance function $d(\omega)$ is given by

$$d(\omega) = \sqrt{tr(\boldsymbol{\Sigma}(\omega_0)^{-1}\boldsymbol{\Sigma}(\omega)) - n - \ln|\boldsymbol{\Sigma}(\omega_0)^{-1}\boldsymbol{\Sigma}(\omega)|}$$

where $\boldsymbol{\Sigma}(\omega) = (1-\omega)\tilde{\boldsymbol{\Sigma}}_1 + \omega\tilde{\boldsymbol{\Sigma}}_2$, and λ is a hyperparameter.

Note that the prior for $\omega_0 = 1$ follows from reversing the roles of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. The approach suggested by Fuglstad et al. (2020) for specifying λ is to choose a value such that the median is $\omega_m = 0.25$ in the case where $\omega_0 \in (0, 1)$ and such that $\mathrm{P}(\mathrm{logit}(\omega) + \mathrm{logit}(1/4) < \mathrm{logit}(\omega) < \mathrm{logit}(\omega_0) + \mathrm{logit}(3/4)) = 1/2$ in the case where ω_0 is the median.

For splits expressing ignorance there are two options. The first is to express ignorance through a series of split priors with base models such that the base case distributes the variance evenly, however, this is cumbersome, and dependent on how we choose to order the splits. The more convenient and intuitive option is to use Dirichlet priors. These have the form

$$\pi(\boldsymbol{\omega}) = \frac{1}{B(\alpha_1, ..., \alpha_K)}\prod_{k=1}^{K}\omega_k^{\alpha_{k-1}}, \ \boldsymbol{\omega} \in \Delta^K$$

where $\Delta_K = \{\boldsymbol{\omega} \in \mathcal{R}^K : \sum_{i=1}^K \omega_i = 1\,, \omega_i > 0 \text{ for } i = 1, 2, ..., K\}$ and B is the multivariate beta function.

In order to express ignorance, the split prior has to be symmetric, so we choose $a_1 = a_2 = ... = a_k = a$. The recommendation regarding specifying this parameters is to set a such that $P(\mathrm{logit}(1/4) < \mathrm{logit}(\omega_1) - \mathrm{logit}(\omega_0) < \mathrm{logit}(3/4)) = 1/2$. By the symmetry of the distribution, this requirement holds for every component of $\boldsymbol{\omega}$ if it's enforced for one component.

Finally, to complete the framework, the question of dependencies between the variance parameters must be answered. Here Fuglstad et al. (2020) postulates two major simplifications in that splits are assumed to be dependent on only their direct descendants, and only through the prior for their base values, not their actual ones. In other words, the model for the entire latent part of the tree structure is

$$\pi(\sigma_1^2, ..., \sigma_N^2) = \pi(t|\{\boldsymbol{\omega}_s\}_{s=1}^S \prod_{s=1}^{S} \pi(\boldsymbol{\omega}_s|\{\boldsymbol{\omega}_j = \boldsymbol{\omega}_j^0\}_{j\in D(s)}) \tag{2.7}$$

where $D(s)$ is the set of direct descendant nodes of split number s, $\boldsymbol{\omega}_j \in \Delta^{l_j}$ where Δ^n denotes the set $\{\boldsymbol{x} \in \mathcal{R}^n \, : \, \sum_{i=1}^n x_i = 1 \, , \, x_i \geq 0 \; \forall i\}$ and l_s is the number of direct descendant splits in split s.

Implementing this in practice can be rather difficult. For more than trivial covariance matrices, the direct way to handle the prior from Theorem 2 during MCMC is to recompute it at every step. Because computing $\pi(\omega)$ involves inverting the resulting covariance matrices, it is computationally infeasible with common random effect dimensions.

Luckily, Hem et al. (2021) has developed an R package, makeMyPrior, that automates the process of defining and fitting HD priors using a numerical approach. It features built-in functions for specifying any variance prior within the HD prior framework, setting up tree structures, choosing whether to express ignorance or some form of knowledge, as well as PC or other variance priors for root and singleton nodes Like the HD prior framework overall, these builtin functions are restricted to random effects. In order to add fixed effects to the tree, we will have to write custom makemyprior Stan code. Luckily, the library also facilitates this.

2.4 The No-U-turn sampler

The No-U-turn sampler(NUTS) is an extension of Hamiltonian Monte-Carlo(HMC), in turn an extension of the general MCMC method, making it an improvement over an algorithm which itself greatly improves upon generic MCMC. The No-U-turn sampler is particularly efficient at approximating samples from joint posteriors with potentially difficult geometry, and will therefore be the workhorse of all inference in this paper. In this section, we will cover the mathematical theory behind the algorithm, and, in Section 2.4.5, briefly discuss Stan, the implementation of NUTS on which we will be relying.

2.4.1 Markov chain Monte Carlo methods

In general, applying Bayesian statistics to realistically complex problems leads to posterior distributions that are not analytically tractable to integrate over. Getting applied Bayesian statistics off the ground has thus been largely dependent on innovations in computers and algorithms that enable researchers to approximate these distributions numerically. The general term for most of these techniques is Markov chain Monte Carlo(MCMC), and the most basic type of MCMC is given by the Metropolis-Hastings algorithm.

As detailed by Givens and Hoeting (2012), the algorithm approximates sampling $\boldsymbol{X}$ from the target distribution $f(\boldsymbol{x})$ by first choosing an initial value $\boldsymbol{x}_0$ such that $f(\boldsymbol{x}_0) > 0$ and then, for every step, $t = 1, 2, ...$ proceeding as follows.

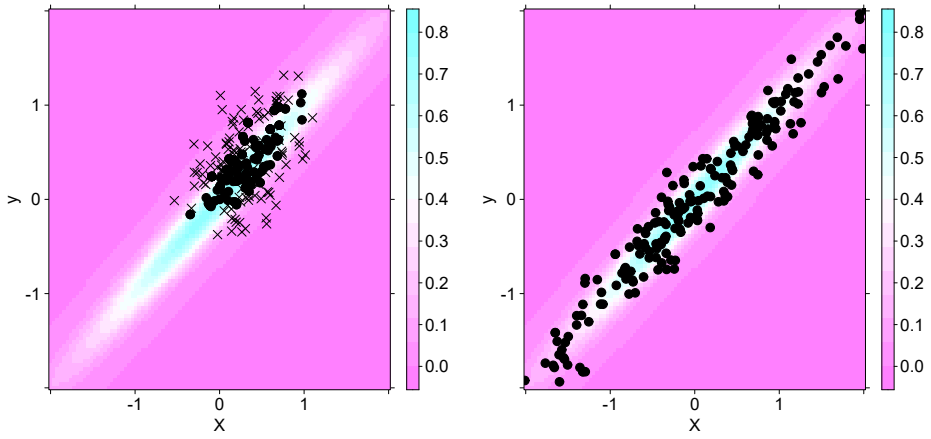
1. Sample $\boldsymbol{x}^* \sim g(\cdot \mid \boldsymbol{x}_t)$
2. Compute $R(\boldsymbol{x}_t, \boldsymbol{x}^*)$, where $R(\boldsymbol{u}, \boldsymbol{v}) = \frac{f(\boldsymbol{v})g(\boldsymbol{u},\boldsymbol{v})}{f(\boldsymbol{u})g(\boldsymbol{v},\boldsymbol{u})}$
3. Accept $\boldsymbol{x}^*$ as the next value $\boldsymbol{x}_{t+1}$ with probability $\min\{R(\boldsymbol{x}_t, \boldsymbol{x}^*), 1\}$. Otherwise set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$.

It can be shown that for reasonable choices of g, the Markov chain defined by this process will converge in distribution to the target distribution, f. For details, see Givens and Hoeting (2012).

While this algorithm is highly flexible and can in principle be applied to any target distribution, it is still far from optimal. For starters, its basic version can exhibit undesireable random walk behavior wherein it struggles to explore distributions with problematic geometry and doubles back on itself, effectively doing work to more or less stay in place.

Example 1. Metropolis Hastings over a strongly correlated normal distribution: As a demonstration, consider the MH algorithm on a distribution from which we know how to sample without using MCMC. Let $\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} 1 & 0.98 \\ 0.98 & 1 \end{bmatrix}$. Figure 2.5 displays the result of running a random walk variant of MCMC wherein the proposal distribution is uniform on a ball with radius 0.5 about the last point in the chain for 200 iterations. As can be seen from the figure, the domain is yet to be adequately explored, and many of the points considered by the algorithm are rejected. The sample generated by NUTS, on the other hand, has clearly performed better, having already explored most of the relevant sample space.

Another problem of the MH-algorithm, and MCMC in general, is tuning. Consider for instance the problem of running MH with a random-walk proposal, with a parameter ϵ controlling how large steps to take. Then the choice of ϵ



(a) MH (b) NUTS

Figure 2.5: A comparison of samples for a highly correlated bivariate normal distribution using MH (2.5a) and NUTS (2.5b). In both cases sample-points are superimposed over a heatmap displaying the true distribution. In both cases, sample points are represented by dots, whereas for the MH sample, the rejected sample points are marked with crosses.

can have substantial consequences for the ensuing inference. If ϵ is set too low, then the algorithm will overall have a very high acceptance probability, but will explore the parameter space very slowly.

On the other hand, if ϵ is set too high, the algorithm will attempt to do steps that are far too large, the overall acceptance probability will drop drastically, and the algorithm will rarely move to new points.

The issue of tuning ϵ and other parameters that affect the sampling process, is generally non-trivial and problem-dependent, but luckily there are extensions of the basic MCMC scheme that seeks to automate this process, and consequently make efficient MCMC schemes more easily available and not restricted to those with the expert knowledge or the time to perform tuning themselves. Two examples of such schemes is Hamiltonian Monte Carlo (HMC), and an extension thereof, NUTS.

2.4.2 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo, Hybrid Monte Carlo, or HMC, improves upon the basic MCMC algorithm by proposing new samples in a manner that utilises the posterior distribution's geometry. The way this is done can be understood as a physical analogy. Let $\boldsymbol{\theta}$ be the vector of parameters to be sampled, of dimension D, distributed according to the probability density function $\pi(\boldsymbol{\theta})$ and define $\mathcal{L}(\boldsymbol{\theta}) = \ln \pi(\boldsymbol{\theta})$. The idea behind HMC is then to conceptualise $\boldsymbol{\theta}$ as the position of a particle in D-dimensional space, introducing an auxiliary parameter vector $\boldsymbol{r} \in \mathbf{R}^D$, and moving through the space in manner that (approximately) conserves the Hamiltonian of this fictitious system. In the simplest case, the components of $\boldsymbol{r}$ are independently standard normally distributed, and so the joint density of the two vectors is $\pi(\boldsymbol{\theta}, \boldsymbol{r}) \propto \exp(\mathcal{L}(\boldsymbol{\theta}) - \frac{\langle \boldsymbol{r}, \boldsymbol{r} \rangle}{2})$, where $\langle \cdot, \cdot \rangle$ is the cross product operator. The Hamiltonian of the the system is then

$$H(\boldsymbol{\theta}, \boldsymbol{r}) = V(\boldsymbol{\theta}) + K(\boldsymbol{r})$$

where $V(\boldsymbol{\theta}) = -\mathcal{L}(\boldsymbol{\theta})$ and $K(\boldsymbol{r}) = \frac{\langle \boldsymbol{r}, \boldsymbol{r} \rangle}{2}$. Updates are performed according to the following differential equations.

$$\frac{d\boldsymbol{\theta}}{dt} = \frac{\partial H}{\partial \boldsymbol{r}}, \quad \frac{d\boldsymbol{r}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\theta}}$$

As with MCMC methods in general, this can generally not be done analytically for systems complex enough to be interesting, and so the updates must be acquired via numerical integration. The leapfrog integrator, a second order quadrature with guaranteed stability for oscillating functions, is chosen as this helps preserve detailed balance, and thus guarantee convergence to the target distribution. This numerical integration is done for a total of L steps at which

point a new $\boldsymbol{r}$ is sampled, and the integration starts over at the newly selected sample parameter space point.

Integrating numerically introduces the issue of tuning the step length ε as well as the number of steps per iteration, L. Without any default rule, these must be chosen by the user, typically requiring tuning runs and sufficient expertise, which lower the overall availability of the algorithm.

HMC also has another drawback in that it will on occasion make "U-turns", where the parameter space trajectory turns back on itself, in which case the integration step ends up spending computing power to move *closer* to its starting point. This is obviously undesirable. Solving both this issue as well as eliminating the need to tune for L and ε is what defines NUTS.

2.4.3 The No-U-turn sampler

The innovation of the No-U-turn sampler is threefold. It adaptively chooses leap-frog step-sizes per iteration, ε_t as well as the corresponding number of leap-frog steps, L_t, and it does this via a stopping condition that simultaneously eliminates the issue of the algorithm taking U-turns.

The criterion for choosing L is where the no-U-turn sampler gets it's name. Using a physical analogy again, we can intuit a naive stopping criterion: For a given leapfrog path where $(\boldsymbol{\theta}, \boldsymbol{r})$ is the initial state and $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{r}})$ is the current state, the algorithm should do no further leap-frog steps when the distance between current and initial position begins to decrease over time, in other words when the following derivative becomes less than zero.

$$\frac{d}{dt}\frac{1}{2}\langle\tilde{\boldsymbol{\theta}}, \boldsymbol{\theta}\rangle = \langle(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}), \frac{d}{dt}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})\rangle = \langle\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}, \tilde{\boldsymbol{r}}\rangle \tag{2.8}$$

This criterion would indeed prevent undesirable random-walk behaviour, but the ensuing proposals would not be guaranteed to converge to the target distribution as they would lack time-reversibility.

To avoid this pitfall, NUTS employs a recursive algorithm that successively step-wise doubles the trajectory, moving either forwards or backwards through fictitious time at every step. This approach can be thought of as building a balanced binary tree, the leaf nodes of which are position-momentum states in parameter space and is illustrated in Figure 2.6. Given this tree, NUTS' stops adding more leap-frog steps to a given iteration when Equation (2.8) is less than zero for the left-most and right-most node of some balanced sub-tree. Once the integration stops, the algorithm deterministically chooses a set $\mathcal{C}$ out of the set $\mathcal{B}$ of all the generated trajectory states such that starting from any state in $\mathcal{C}$ has an equal chance of generating $\mathcal{B}$.

A second stopping criterion is also in place to stop integrating if the error becomes too large. This is enforced by stopping if

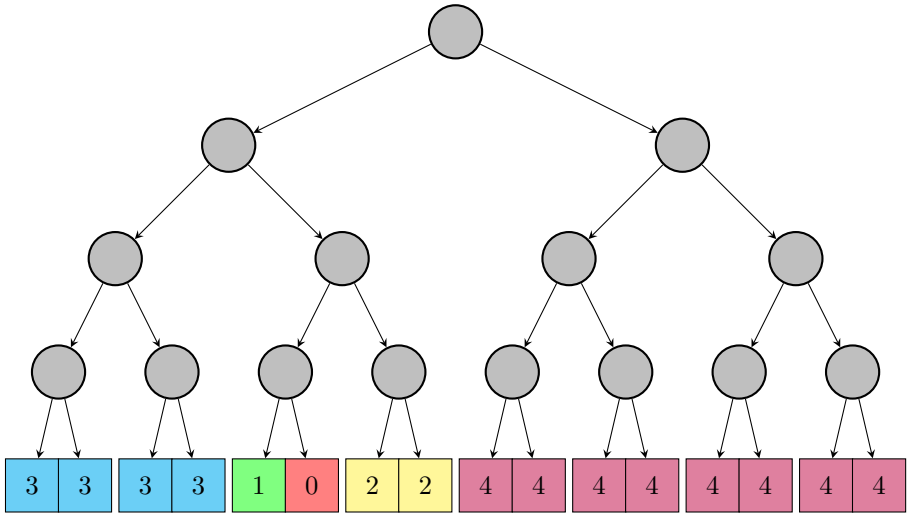


Figure 2.6: An example of a binary tree representing a NUTS integration step trajectory. Labels and colours denote the order in which nodes are added, and the ordering from left to right denotes time-order in the fictitious physical system. A balanced subtree is any set of leaf nodes that share a common ancestor (grey node). In other words all nodes added in the third doubling make up a balanced subtree, as do the ones added during the final one.

$$\mathcal{L}(\boldsymbol{\theta}) - \frac{1}{2}\langle \boldsymbol{r}, \boldsymbol{r} \rangle - \ln u < -\Delta_{max} \tag{2.9}$$

for some leaf node state, where Δ_{max} is some large non-negative number, for instance 10^3 and $u \sim U(0, \exp(\mathcal{L}(\boldsymbol{\theta}) - \langle \boldsymbol{r}, \boldsymbol{r} \rangle))$ is a slicing variable used in the NUTS implementation.

2.4.4 Adaptively choosing ε for NUTS

Hoffman and Gelman (2011) chooses ε adaptively by means of dual averaging. In general, dual averaging is a method for tuning parameters for MCMC at every iteration. Given T iterations, a set of statistics $H_{t\,i=1}^{T}$ providing some information about a specific aspect of the MCMC behaviour at their respective iterations with expectation $h(x) = \mathrm{E}_t[H_t \mid x] = \lim_{T \longrightarrow \infty} \Sigma_{t=1}^{T} \mathrm{E}_t[H_t \mid x_t]$, where $x_t \in \mathbf{R}$ for $t = 1, 2, ..., T$ is a set of tuning parameters, dual averaging performs a series of updates such that the average expectation of H_t approaches zero.

In the particular dual averaging method chosen by Hoffman the tuning parameters, $x_t = \ln \varepsilon_t$ are updated according to

$$x_{t+1} = \mu - \frac{\sqrt{t}}{\gamma(t + t_0)} \Sigma_{i=1}^{t} H_i \quad ; \quad \bar{x}_{t+1} = \eta_t x_{t+1} + (1 - \eta_t)\bar{x}_t$$

where μ is a freely chosen parameter towards which the values x_t are shrunk, γ is a parameter controlling the degree of shrinkage, t_0 is a free parameter that serves to stabilise the initial iterations, $\eta_t = t^{\kappa}$, where $\kappa = 0.75$ and $\bar{x}_1 = x_1$.

As mentioned previously, these updates cause the average expectation of H_t to approach zero. Therefore, by defining $H_t = \delta - H_t^{NUTS}$, where H_t^{NUTS} is the average Metropolis-Hastings acceptance probability of $(\boldsymbol{\theta}, \boldsymbol{r})$ at every node in the tree $\mathbf{B}_t$, Hoffman can control the achieve the desired acceptance rates, and by extension the desired step lengths, by setting δ to be the desired average acceptance probability.

All that remains to fully specify the scheme is to initialise ε_1. Hoffman and Gelman (2011) suggests doing this by starting at $\varepsilon = 1$ and successively doubling or halving the value until the corresponding proposal density according to another MCMC scheme based on a physical analogy, the Langevin proposal, crosses 0.5.

For further details, see Hoffman and Gelman (2011).

2.4.5 Stan

Stan is a probabilistic programming language based on NUTS, and will be the main inferential tool of this paper. Stan is highly flexible, and can more or less run on any model that can be mathematically specified. However, its performance

does depend somewhat on parameterisation. As mentioned in Section 2.4, the integration step will halt, perhaps prematurely, if the criteria in Equation 2.9 occurs, in which case Stan will report a divergent transition. This is particularly likely if model parameters are defined over some closed interval, as crossing the boundaries of these intervals will cause the target likelihood to drop towards negative infinity. If this occurs during warmup, Stan will compensate by making the average step length extra small, causing sampling to be less effective.

To avoid this pitfall it is often best not to specify target distributions directly, but rather use parameter transformations that are in stead defined over the entire real line. For instance, variance parameters are sampled using log variance, and the variance proportions in HD priors discussed in Section 2.3.2 are transformed to logit scale. After sampling on these alternative scale, Stan can easily return the actual desired parameters as via a transformed parameters block and use them as normal.

Inference is also affected by parameters being "coupled", or correlated. If one parameter limits the likelihood of another, such as when a set of parameters control an aspect of the model together, then this will tend to make traversing the parameter space more difficult. To prevent this issue it is best to re-parameterise such that parameters are "de-coupled" as much as possible. The most prominent example of this is to parameterise the RW2 component as a (log) variance parameter and a normalised RW2 vector, $\boldsymbol{u}/\sigma_{RW}$, so that the vector only controls the shape of the non-linear effect, and its amplitude is only controlled by the variance parameter. This is similar to advice given by Gelman (2006).

2.5 Scoring rules

We will need an objective criterion for discriminating between models. Mean square error, or MSE, is a well-known measure of model accuracy, but is somewhat lacking in this context, as it only considers the value(s) of a given prediction, not the associated *uncertainty*.

In general, a function S used to evaluate a model F given the observation Z is called a scoring rule, and we say that it is proper if

$$\mathrm{E}[S(Z, F)] \leq \mathrm{E}[S(Z, G)]$$

in the case where $Z \sim F$.

Despite it's prevalent use, MSE is not a proper scoring rule. Another scoring rule, which is proper and also factors in the uncertainty associated with statistical estimation, is the continuous rank probability score, or CRPS. It is defined thus

$$\mathrm{CRPS}(y, F) = \int_{\mathbf{R}} (F(z) - \mathbf{1}\{z \geq y\})^2 dz$$

where $\mathbf{1}$ is the indicator function. If the observation is a random vector $\boldsymbol{Z}$ the CRPS is simply defined as the average value over all components of said vector. In other words we have

$$\mathrm{CRPS}(\boldsymbol{y}, F) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{CRPS}(y_i, F)$$

where $\boldsymbol{y} \in \mathbf{R}^n$.

Note that according Gneiting and Raftery (2007), the CRPS is defined with the opposite sign, but we choose to define it to be positive instead to provide a more natural interpretation and an comparison with other performance measures.

Example 1. CRPS of a point predictive distribution.

One interesting property of CRPS is that it can be viewed as a generalisation of the mean absolute error. To see this, consider a point prediction-distributions that places all the probability mass at $y = \hat{y}$. The cumulative density function of such a distribution is a step function, and so we obtain

$$\mathrm{CRPS}(y, F) = \int (\mathbf{1}_{y>\hat{y}} - \mathbf{1}_{z>y})^2 dz = \int_{\min(\hat{y},y)}^{\max(y,\hat{y})} dz = \mid y - \hat{y} \mid = \mathrm{MAE}$$

where MAE denotes the mean absolute error. This also demonstrates that CRPS generalises beyond statistical models to deterministic approaches, such as neural nets and other types of machine learning, or single meteorological models.

Example 2. Using CRPS to differentiate between models with equal prediction-observation distance

To illustrate how this scoring rule compares to MSE, consider evaluating the predictive distributions $\mathcal{N}(0, 0.3)$, $\mathcal{N}(0, 3)$ and $\mathcal{N}(2, 1)$ given the observation $Y = 1$. These distributions are displayed in Figure 2.7. From inspecting the plot, one would expect the third distribution to be best by a small margin, but because all three distributions have modes that equidistant from $Y = 1$, they are all equal with respect to MSE.

CRPS, on the other hand, does recognise this slight difference, and does indeed indicate that the third predictive distribution is the one that best fits the observation. Although analytical solutions to the integral generally are not easily available, there is one for the normal distribution. If the predictive distribution is $\mathcal{N}(\mu, \sigma^2)$, and the observation is y then we have

$$\mathrm{CRPS}(\mathcal{N}(\mu, \sigma^2)) = \sigma[\pi^{-1/2} - 2\phi(z) - z(2\Phi(z) - 1)]$$

where we have defined $z = \frac{y-\mu}{\sigma}$ and ϕ and Φ is the standard normal probability function and cumulative density function, respectively.

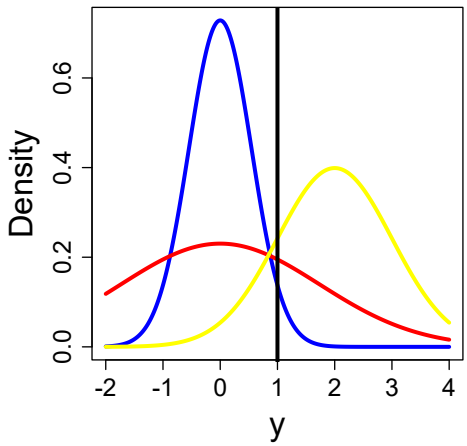


Figure 2.7: A comparison of the predictive distributions in Example 2 in Section 2.5, with a vertical line representing the observation, and the blue, red and yellow curves representing the probability density functions of the $\mathcal{N}(0, 0.3)$, $\mathcal{N}(0, 3)$ and $\mathcal{N}(2, 1)$ distributions, respectively.

The ensuing values for the distributions we consider are then displayed in Table 2.1

(μ, σ^2)	MSE	CRPS
$(0, 0.3)$	1	0.70
$(0, 2)$	1	0.629
$(2, 1)$	1	0.602

Table 2.1: A comparison of CRPS and MSE values for a selection of predictive distributions $\mathcal{N}(\mu, \sigma^2)$ given the observation $y = 1$.

Chapter 3

HD variance priors for non-linear smoothing

In Example 1 in Section 2.1, we laid out the general model to be used in this paper, a p-dimensional non-linear smoothing model. Here, we demonstrate said model, as well as the current HD prior framework with a basic example: Modelling a noisy non-linear signal in one dimension. We will also be comparing the HD priors to an independent prior setup, in terms of both objective performance scores and how prior knowledge is specified and encoded, and what implications this appears to have for the ensuing posteriors. These implications seem to become most apparent in Section 3.2, where we compare how the priors perform when they are misspecified.

The data for this section will be generated as the sum of a simple linear function, a sine wave, and Gaussian noise:

$$y_n \sim \mathcal{N}(\alpha + n\beta + \sin\frac{2\pi n}{30}, \sigma_R^2)\,,\ n = 1, 2, ..., N \tag{3.1}$$

with $N = 50$, $\alpha = -1/10$, $\beta = 1/10$ and $\sigma_R = \sqrt{0.1}$. As can be seen from Figure 3.1, data generated from this distribution clearly exhibits both linear and non-linear behavior, as the function oscillates sinusoidally around a linear trend.

3.1 Model fits using existing priors

We will model this signal with three different joint variance priors, but before introducing these we will explain the rest of the model. In all three cases the model has the same basic likelihood

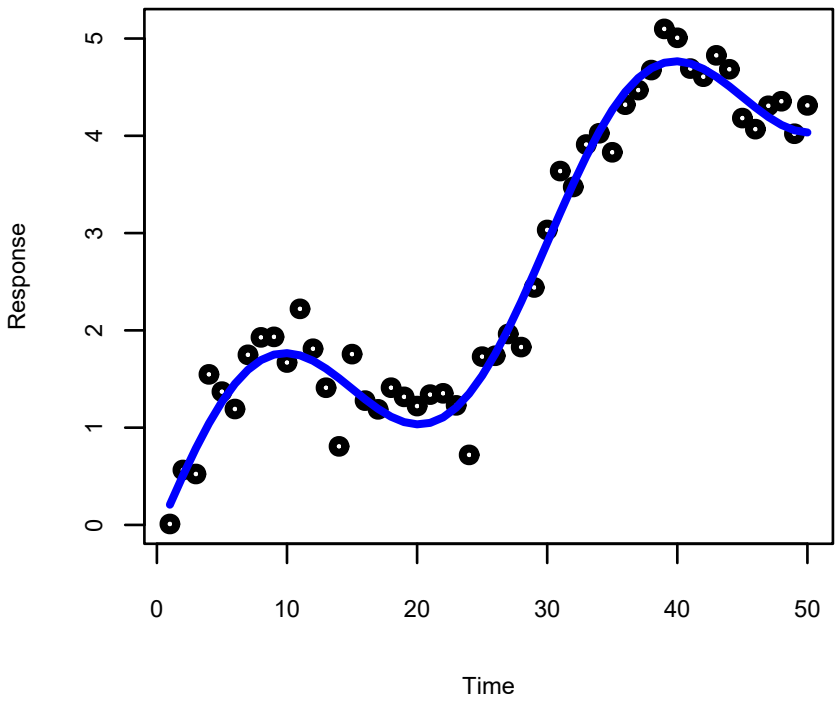


Figure 3.1: The data distribution in this chapter. The dots display observed values from the distribution in Equation 3.1, whilst the blue line represents the signal without any error, or in other words the expectation of the distribution in Equation 3.1.

$$y_n \mid \alpha, \beta, \boldsymbol{u}, \sigma_R \sim \mathcal{N}(\alpha + n\beta + u_n + \epsilon_n, \sigma_R)\,,\ n = 1, 2, ..., N \tag{3.2}$$

where $\boldsymbol{u}$ is a vector of parameters used to capture the signal's non-linear behavior by means of a second order random walk model, as discussed in Example 2. The variance for the forward differences that define $\boldsymbol{u}$ is $\sigma^2_{\boldsymbol{u}}$.

The ensuing second-order IGMRF can be made proper by introducing two constraints. We do so by specifying that $\boldsymbol{u}$ is orthogonal to the intercept and linear trends.

$$\bar{\boldsymbol{u}} = \frac{1}{N}\sum_{n=1}^{N} u_n = 0 \quad \text{and} \quad \frac{\sum_{n=1}^{N} u_n(n - \frac{N+1}{2})}{\sum_{n=1}^{N}(n - \frac{N+1}{2})^2} = 0 \tag{3.3}$$

The last two basic model components besides the variances are the priors for α and β, namely $\alpha \sim \mathcal{N}(0, 30)$ and $\beta \sim \mathcal{N}(0, 100)$.

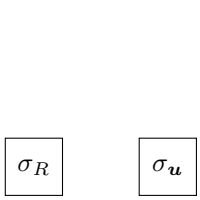
For the variance priors we test three different approaches. First we fit a model with independent PC priors on σ_R and $\sigma_{\boldsymbol{u}}$, then we test a HD prior with an ignorance prior for $\omega = \frac{\sigma^2_{\boldsymbol{u}}}{\sigma^2_{\boldsymbol{u}} + \sigma^2_R}$ and a PC prior on the total model variance, V, and finally a HD prior with a PC on ω in addition to the one for V. The priors are illustrated in Figure 3.2. In general, we try to encode the prior belief that $V = 1$, and $\omega = 0.25$. To see the reason for the former, note that, because $\pi(\beta)$ is not included in the HD prior, it only needs to explain the variance that remains after removing the linear trend. From inspecting Figure 3.3b, we see that non-linear part of the data mostly span the interval $(-1, 1)$, hence the prior belief for V. The value for ω, meanwhile, is chosen to provide shrinkage towards random noise whilst still enabling the model to capture non-linear behaviour given sufficiently strong evidence.

It is worth noting at this point that we scale $\boldsymbol{u}$ in our implementation such that it does not exhibit extremely large marginal variances. We do this by choosing a "representative value" of the random walk variance in the same manner as Sørbye and Rue (2014), namely the geometric mean of the diagonal entries of the generalised inverse of $\boldsymbol{u}$'s improper precision matrix, and dividing all the vector entries by this value. This means the exact values of the marginal variances of the components of $\boldsymbol{u}$ does not need to be factored in to the process of specifying prior knowledge. This helps keep it intuitive, in line with Bayesian workflow(Gelman et al., 2020).

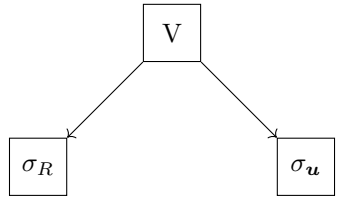
Our comparison shows that HD priors perform equally well or better than the "basic" prior model, and also highlights some other benefits of the HD approach.

We collect statistics summarising the models' performance in Table 3.1.

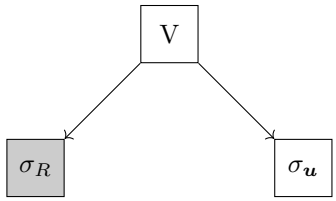
Example 1. Independent priors: Our first model is the simplest, and ironically the most involved when it comes to specifying prior knowledge, as we must first "translate" our prior belief in terms of independent prior parameters.



(a) The HD tree structure for the independent prior.



(b) The structure of the ignorant split HD prior.



(c) The structure of the PC split HD prior.

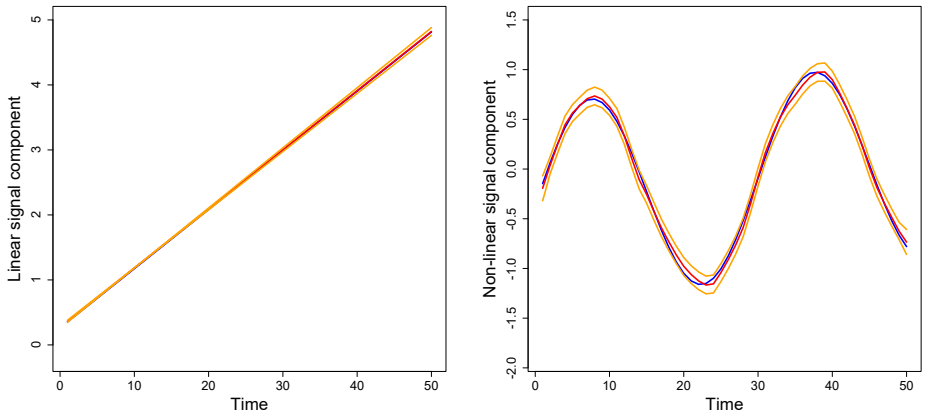
Figure 3.2: HD structures for the variance priors in this section. The graph in Figure 3.2a is entirely disconnected as no decomposition is in effect yet. The trees in 3.2b and 3.2c have the simplest possible structure and differ only in latter's preference towards σ_R.

We assign independent PC priors for σ_R and $\sigma_{\boldsymbol{u}}$. In other words, the priors are as described in Section 2.3.1, $\sigma_R \sim \exp(\lambda_R)$ and $\sigma_{\boldsymbol{u}} \sim \exp(\lambda_{RW})$. The hyperparameters are then set such that the ensuing a priori medians are the same as the total variance V multiplied by the variance split ω would be for the HD priors. In other words, we choose parameters such that the prior for σ_R has median $0.75V$, and the one for $\sigma_{\boldsymbol{u}}$ has median $0.25V$.

Decomposing the signal and estimate in their linear and non-linear components shows that both seem to be modeled quite well, as can be seen in Figure 3.3.

Example 2. Joint variance prior with ignorance split: Here we reparameterise as described in Section 2.3.2, so that we work with ω and V instead of σ_r and $\sigma_{\boldsymbol{u}}$. The prior for V is assigned in the same manner as for the independent variances in Example 1, with an exponential prior on the standard deviation with a priori correct median, and because we only have a single split, the prior for ω is simply $\omega \sim u[0, 1]$. We do not include estimate plots for this example, nor for Example 3, as both appear only negligibly different from Figure 3.3.

Example 3. HD prior with PC split: In this model, we keep the reparam-



(a) Linear signal component with corresponding confidence interval

(b) Non-linear signal component with corresponding confidence interval

Figure 3.3: Comparisons of the true signal components, their corresponding estimates and 95 % credibility intervals given the model in Equation (3.2) and independent priors. True signal components are represented with a blue line, red signifies our estimate, and the orange curves mark 95% confidence intervals.

Table 3.1: Performance statistics for the priors in Chapter 3.

Prior(s)	**MSE** (10^{-3})	$\overline{(\eta_{0.975} - \eta_{0.025})}(10^{-1})$	**CRPS** (10^{-2})
Independent	1.69	3.18	3.33
Ignorance HD	1.55	3.17	3.47
PC HD	1.64	3.15	3.38

eterisation from Example 2, but use the PC prior for ω described in Theorem 2. Setting hyperparameters is then straightforward, we simply set hyperparameters for $\pi(V)$ and $\pi(\omega)$ such that the medians match our prior beliefs. Again, the model performs on par with previous setups.

As we can see, the priors seem to perform more or less on par based on this example. All margins of error are so small that it is not certain what part of the fluctuations are due to the priors per se, and which are randomness from the Stan-inference. In any case this example suggests that HD priors perform on par with competing priors, if not better.

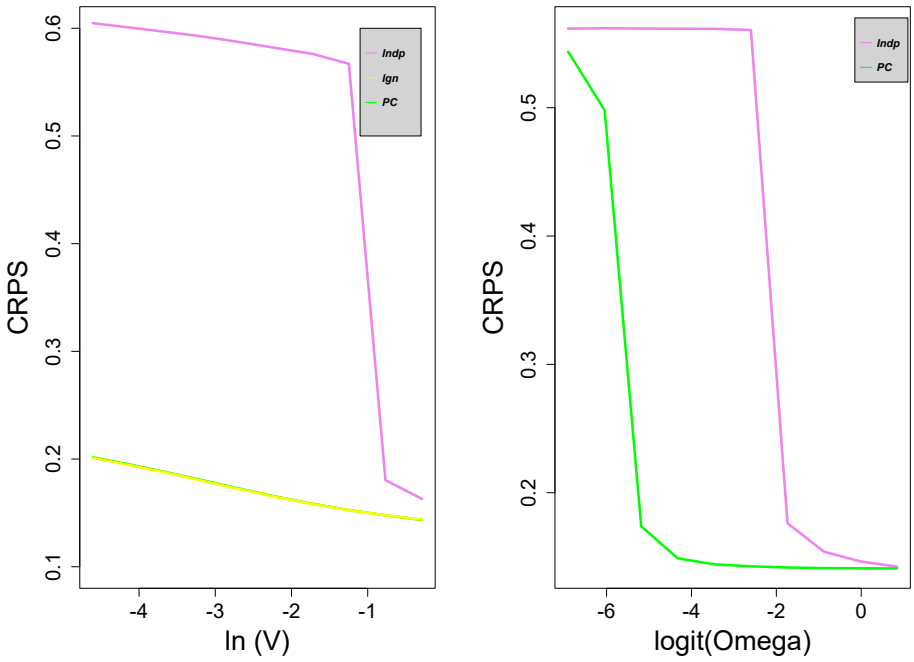
3.2 Model behaviour under prior misspecification

We have seen that all three priors lead to good model performance given reasonable specification. Here we will compare how they behave when mis-specified. To do so, we will vary the belief encoded into each prior, and compute the ensuing CRPS, fixing ω and varying V, and then vice versa. Recall that we set $V = 1$ and $\omega = 0.25$ in the base case. Increasing V beyond 1 is uninteresting, as there is enough data to pull the model towards more reasonable values of V for any realistic over-specification of V. On a similar note, our permissible a priori beliefs for ω are confined to $(0, \sqrt{1/2})$. To see why, recall that $\pi(\omega)$ is given in Theorem 2 and note that the value of λ constrains the range of permissible median values of ω. In the liminal case where $\lambda \to 0$, all the mass is concentrated around $\omega = 0$ and for the opposite case we see that $\lambda \to \infty$ leads to $\pi(\omega) \to \frac{1}{2\sqrt{\omega}}$. The median of this liminal distribution is $\sqrt{1/2} \approx 0.707$, and so it does not make sense to consider values of ω outside of $(0, 0.7)$, unless we want to reverse the roles of $\boldsymbol{u}$ and $\boldsymbol{\epsilon}$ in the prior and have shrinkage towards non-linearity, which would be against the principle of parsimony. We also want to limit ourselves to mis-specifications that are interesting without seeming too unrealistic, so for V we look at a prior expectations in the interval $(0.01, 0.75)$. The results can be

seen in figures 3.4a and 3.4b. Note that the comparison in Figure 3.4b is not applicable to the ignorance prior, hence why it only features in the plot where we vary the specified belief about the variance.

Clearly it appears the HD priors are more robust under mis-specification, either taking much more extreme values before getting significantly worse, as we can see in Figure 3.4b, or not reaching this point at all, as we see in figure 3.4a, in stead worsening at a steady but slow pace.

This is, in part, to be expected given how HD priors are implemented. Specifying medians for V and ω is a more "soft" way of encoding prior knowledge than setting medians for each component separately, as we can see in how the different priors behave in Figure 3.4b. The difference is even more dramatic when misspecifying V. We suspect this too is because of how ω is encoded in the different priors. Recall that we are fixing the priors at $\omega = 0.25$ when varying V, and that the way this is encoded for the independent priors is by proportioning the medians of $\pi(\sigma_R)$ and $\pi(\sigma_{\boldsymbol{u}})$. This, combined with the model having less ability to stray from specified variance proportions likely means that as $V \to 0$, $\pi(\sigma_{\boldsymbol{u}})$ is having all its probability mass moved towards 0 at a faster rate for the independent prior than for the HD one, forcing the model to explain all non-linear variance as residual variance and thus drastically worsening model performance earlier.



(a) Plot of CRPS values to $\ln(V)$.

(b) Plot of CRPS values to $\text{logit}(\omega)$.

Figure 3.4: Plots of CRPS values under mis-specification. Independent, ignorant and PC HD priors are represented with a purple, yellow and green curve respectively. Transformations of V and ω are used to better highlight interesting features.

Chapter 4

Adding fixed effects to the HD prior

As a first step towards more complex modelling, we incorporate the variance associated with the linear coefficient into the HD tree, as shown in Figure 4.1. Note that we will not be including the variance from the intercept, as imposing shrinkage on the intercept seems generally redundant.

4.1 Adding linear effect variance to the variance hierarchy

When Fuglstad et al. (2020) first formulated the HD prior framework, they noted that they would be focusing on random effects. As such, the HD prior framework can currently only handle random effects. Despite this, the prior for this new case actually follows quite easily.

Theorem 3. *Let $\boldsymbol{y} \in \mathbf{R}^N$ be modelled according to Equation (2.1), and let the contributions from $\boldsymbol{u}_i$ and β_j, for $i, j \in \{1, 2, ..., p\}$, enter the model through $\boldsymbol{A}_{\boldsymbol{u}_i} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\boldsymbol{u}_i} \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_i})$ and $\beta_j \boldsymbol{x}_j \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\beta_j} \boldsymbol{x}_j^T \boldsymbol{x}_j)$, where the covariate vectors have been standardised such that $\|\boldsymbol{x}_j\| = 1$ and $\sum_{i=1}^N x_{ji} = 0$ for $j = 1, 2, ..., p$.*

Then the prior for the split ω between $\boldsymbol{u}_i$ and β_j has the following expression.

$$\pi(\omega) = \frac{\lambda \exp(-\lambda\sqrt{\omega})}{2\sqrt{\omega}(1 - \exp(-\lambda))}$$

Proof. The proof proceeds similarly to the corresponding proof in the supplementary material of Fuglstad et al. (2020) et al., with a few key differences.

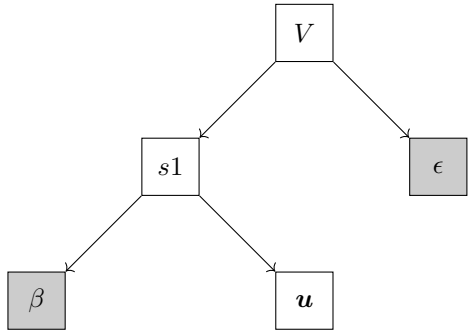


Figure 4.1: The HD tree incorporating the variance associated with a single linear effect β into the hierarchy.

For simplicity, we simply refer to $\boldsymbol{u}_i$ and $\boldsymbol{x}_j$ as $\boldsymbol{u}$ and $\boldsymbol{x}$ for the remainder of the proof.

The proof in the supplementary material begins by noting that for random effects $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ entering the linear predictor through $\boldsymbol{A}_i\boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, \sigma_i^2\tilde{\boldsymbol{\Sigma}}_i)$, $i = 1, 2$ both $\tilde{\boldsymbol{\Sigma}}_1$ and $\tilde{\boldsymbol{\Sigma}}_2$ are positive semi-definite and their sum is invertible. The matrices are positive semi-definite for our case as well, but it does not follow that the sum corresponding to $\tilde{\boldsymbol{\Sigma}}_1 + \tilde{\boldsymbol{\Sigma}}_2$ is non-singular, as both covariance matrices have zero-valued eigenvalues.

More specifically, $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}}$ has rank deficiency 2, $\boldsymbol{x}^T\boldsymbol{x}$ has rank deficiency $N - 1$, and the sum has rank deficiency 1, as adding $\boldsymbol{x}^T\boldsymbol{x}$ to $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}}$ introduces $\boldsymbol{x}$ as a non-zero eigenvector.

In any case, the resulting sum is still singular. We can however work around this by simply omitting the remaining one dimension which the sum does not span and then proceeding as in the original proof.

We can now define $\boldsymbol{\Sigma}'(\omega) = \omega\boldsymbol{\Sigma}'_{\boldsymbol{u}} + (1 + \omega)\boldsymbol{\Sigma}'_\beta$ where $\boldsymbol{\Sigma}'_\beta$ and $\boldsymbol{\Sigma}'_{\boldsymbol{u}}$ are the projections of $\boldsymbol{x}^T\boldsymbol{x}$ and $\boldsymbol{\Sigma}_{\boldsymbol{u}}$ onto the span of $\boldsymbol{\Sigma}_{\boldsymbol{u}}$, respectively. $\boldsymbol{\Sigma}'(\omega)$ is now invertible, and $\boldsymbol{\Sigma}_\beta$ is still singular, that means that the distance function for the case in which $\boldsymbol{\Sigma}_\beta$ is invertible, and the distance function

$$d(\omega) = \sqrt{\text{tr}(\tilde{\boldsymbol{\Sigma}}_1^{-1}\boldsymbol{\Sigma}(\omega)) - n - \ln \mid \tilde{\boldsymbol{\Sigma}}_1^{-1}\boldsymbol{\Sigma}(\omega) \mid}$$

goes to infinity for every non-zero value of ω, so we instead define the distance as the liminal value of $d(\omega; \omega_0)$ as $\omega_0 \to 0$, because $d(\omega; \omega_0)$ is finite for any non-zero ω_0.

All that remains then is to define $\lambda(\omega_0)$ such that $d(\omega; \omega_0)$ converges to a finite value as $\omega_0 \to 0^+$.

Prior(s)	**MSE** (10^{-3})	$\overline{(\eta_{0.975} - \eta_{0.025})}(10^{-1})$	**CRPS** (10^{-2})
Independent	1.69	3.18	3.33
Ignorance HD	1.55	3.17	3.47
PC HD	1.64	3.15	3.38
Expanded PC HD	1.62	1.89	2.34

Table 4.1: Performance statistics for all the variance priors so far

As in Fuglstad et al. (2020)'s original proof, we define the matrix $\boldsymbol{P}$ such that $\boldsymbol{P}(\boldsymbol{\Sigma}'_{\boldsymbol{u}} + \boldsymbol{\Sigma}'_{\beta})\boldsymbol{P}^T = \boldsymbol{I}$, and $\boldsymbol{S}_i = \boldsymbol{P}\boldsymbol{\Sigma}'_i\boldsymbol{P}^T$, where $\boldsymbol{\Sigma}'_1 = \boldsymbol{\Sigma}'_{\boldsymbol{u}}$ and $\boldsymbol{\Sigma}'_2 = \boldsymbol{\Sigma}'_{\beta}$.

We then get an alternate distance function given by

$$d(\omega;\omega_0)^2 = \text{tr}(\boldsymbol{S}(\omega_0)^{-1}\boldsymbol{S}(\omega)) - n - \ln \mid \boldsymbol{S}(\omega_0)^{-1}\boldsymbol{S}(\omega) \mid$$

where $\boldsymbol{S}(\omega) = (1-\omega)\boldsymbol{S}_1 + \omega\boldsymbol{S}_2 = \omega\boldsymbol{I} + (1-2\omega)\boldsymbol{S}_1$ because $\boldsymbol{S}_1 + \boldsymbol{S}_2 = \boldsymbol{I}$.

The distance can then be computed by writing $\boldsymbol{S}_1 = \Sigma_{i=1}^{N}[(1-2\omega)\lambda_i + \omega]\boldsymbol{v}_i\boldsymbol{v}_i^T$, where λ_i and $\boldsymbol{v}_i$ is the ith eigenvalue and eigenvector of $\boldsymbol{S}_1$, respectively.

Because $\boldsymbol{S}_1$ has rank deficiency $N-1$, this degenerates to a single term corresponding to the single non-zero eigenvalue of $\boldsymbol{S}_1$. If we assume the eigenvalues are sorted in decreasing order we have the following.

$$d(\omega;\omega_0) = (N-1)(\frac{\omega}{\omega_0} - \ln\frac{\omega}{\omega_0}) + \frac{(1-2\omega)\lambda_1 + \omega}{(1-2\omega_0)\lambda_1 + \omega_0}$$

Now we need only introduce a new scaled distance, $\tilde{d}(\omega;\omega_0) = \omega_0 d(\omega;\omega_0)$ and an expression expression for λ that makes the limit converge, $\lambda(\omega_0) = \sqrt{\omega_0/(N-1)}\tilde{\lambda}$ and the distribution function follows by taking the limit.

$$\pi(\omega) = \frac{\tilde{\lambda}\exp(\tilde{\lambda}\sqrt{\omega})}{2\sqrt{\omega}(1-\exp(-\tilde{\lambda}))}\,,\ \omega \in (0,1)$$

□

The performance statistics of this new prior are shown and compared to those of the previous priors form Chapter 3 in Table 4.1

As we can see, the new prior performs quite well, scoring second best out of all the HD priors when it comes to MSE, and outperforming every other prior by a relatively significant margin when it comes to credibility interval width and CRPS. From this example it then seems that the new prior offers similarly accurate predictions, but with a significant increase in accuracy.

4.2 Redefining the total variance for the expanded HD prior

This expansion of the HD prior framework calls for a re-evaluation of what we consider the "total" variance in a split or root node, as Fuglstad et al. (2020) only ever considered decomposing entirely random effects, not linear coefficients. Though it might seem intuitive to simply take the sum total, or the average of $\text{var}(\boldsymbol{y})$, conditional on all the model parameters, we find that the choice most in line with Fuglstad et al. (2020)'s original ideas, and also the one yielding the most intuitive results, is taking the "average" of the trace of $\tilde{\boldsymbol{\Sigma}}_V$, which we have defined in a manner analogous to the covariance matrices in Theorem 2. More specifically, it is the sum of all the covariance structure matrices from all the model components, weighted by their corresponding component variances. So for the model in Section 3 we have

$$\tilde{\boldsymbol{\Sigma}}_V = \omega_\beta \sigma_\beta^2 \boldsymbol{x}\boldsymbol{x}^T + \omega_{\boldsymbol{u}}^2 \sigma_{\boldsymbol{u}}^2 \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}} + \omega_R \sigma_R^2 \boldsymbol{I}$$

where the structure matrix for $\boldsymbol{u}$, $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}}$, is as explained in Example 2 in Section 2.2.1. Also note that we diverge from the usual split notation in this section and the remainder of this chapter by letting the usual symbols for the leaf node split weights refer to the de facto portions of the total variance assigned to their respective components, and not merely the portion they are assigned from their (non root) parent node. So for instance we write $\omega_{\boldsymbol{u}}$ in stead of the arguably more correct alternative $\omega_{S1}\omega_{\boldsymbol{u}}$.

The "average" variance is then taken by dividing by N.

$$V = \frac{1}{N}\text{tr}(\tilde{\boldsymbol{\Sigma}}_V) = \frac{1}{N}\text{tr}(\omega_\beta \sigma_\beta^2 \boldsymbol{x}^T\boldsymbol{x} + \omega_{\boldsymbol{u}}^2 \sigma_{\boldsymbol{u}}^2 \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}} + \omega_R \sigma_R^2 \boldsymbol{I})$$

Because the trace function is linear, this is the same as the weighted sum of the traces from each structure matrix.

Here we will make a quick note about the term corresponding to σ_β^2. Recall from Section 4.1 that we standardise covariate vector such that it sums to 0 and has unit norm. This important for keeping the role of σ_β^2 clear, as makes the contribution from β simply $\omega_\beta\sigma_\beta^2$ whereas it would otherwise be a function of the covariate vector, which would be less intuitive, and thus undesirable.

The contribution from the residual error has the same form without any extra work, as the assumption of identical independence means $\tilde{\boldsymbol{\Sigma}}_R = \boldsymbol{I}_p$ and so $\text{Tr}(\tilde{\boldsymbol{\Sigma}}_R) = N$. We then get the following total variance.

$$V = \omega_\beta \sigma_\beta^2 + \frac{1}{N}\omega_{\boldsymbol{u}}\sigma_{\boldsymbol{u}}^2 \text{tr}(\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}}) + \omega_R \sigma_R^2$$

4.3 Extending the new total variance to multiple covariates

Extending the notion of total variance to the multivariate case is not much more complicated. Consider the model with likelihood

$$\boldsymbol{y}|\alpha, \boldsymbol{\beta}, \boldsymbol{u}, \sigma_R \sim \mathcal{N}(\alpha + \sum_{i=1}^{p} \beta_i \boldsymbol{x}_i + \boldsymbol{u}, \sigma_R^2) \tag{4.1}$$

in other words, the same model as in Section 4.2, but with the addition of any number of additional covariates, each with a independently identically distributed linear effect.

Again, standardising the covariates simplifies things significantly, making it so $\langle \boldsymbol{X}_i, \boldsymbol{X}_j \rangle = \delta_{ij}$, which implies $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} = \boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{I}_p$. This means we then get an analogous result to the one in Section 4.2

$$\mathrm{tr}(\omega_{\boldsymbol{\beta}} \sigma_{\boldsymbol{\beta}}^2 \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}}) \frac{1}{N} = \omega_{\boldsymbol{\beta}} \sigma_{\boldsymbol{\beta}}^2 N \frac{1}{N} = \omega_{\boldsymbol{\beta}} \sigma_{\boldsymbol{\beta}}^2$$

and the total variance follows quite simply once more.

$$V = \omega_{\boldsymbol{\beta}} \sigma_{\boldsymbol{\beta}}^2 + \omega_{\boldsymbol{u}} \sigma_{\boldsymbol{u}}^2 \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}}) \frac{1}{N} + \omega_R \sigma_R^2$$

4.4 Extending the new total variance to the general case

As a next step, consider a general model with an arbitrary number S of random effects, besides the residual. This means the likelihood is then

$$\boldsymbol{y}|\alpha, \boldsymbol{\beta}, \sigma_R^2, \{\boldsymbol{u}_s\}_{s=1}^S \sim \mathcal{N}(\alpha + \boldsymbol{X}\boldsymbol{\beta} + \sum_{s=1}^{S} \boldsymbol{u}_s, \sigma_R^2) \tag{4.2}$$

where $\boldsymbol{u}_s$ is the contribution from random effect s to the linear predictor. The total covariance matrix is then

$$\tilde{\boldsymbol{\Sigma}}_V = \omega_{\boldsymbol{\beta}} \sigma_{\boldsymbol{\beta}}^2 \tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\beta}} + \sum_{i=1}^{S} [\omega_i \sigma_i^2 \tilde{\boldsymbol{\Sigma}}_i] + \omega_R \sigma_R^2 \boldsymbol{I}_p$$

after which the total variance follows rather straightforwardly.

$$V = \omega_{\boldsymbol{\beta}} \sigma_{\boldsymbol{\beta}}^2 + \frac{1}{N} \Sigma_{i=1}^{S} \omega_i \sigma_i^2 \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}) + \omega_R \sigma_R^2$$

4.5 Performing shrinkage between covariates

Enabling the model to select between contributions from specific covariates, or sets of covariates might be another natural extension of the framework. Consider again the model given by (4.2), but now focus only on the total linear coefficient variance. If we want the model to select between two sets of covariates $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k\}$ and $\{\boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+2}, ..., \boldsymbol{x}_p\}$ with corresponding design matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ respectively, both standardised as in Section 4.3, then we can let the model select between them by introducing $\omega_{\boldsymbol{\beta}_1}$ and $\omega_{\boldsymbol{\beta}_2}$ such that

$$\omega_\beta \tilde{\boldsymbol{\Sigma}}_\beta = \sigma^2_{\beta_1} \tilde{\boldsymbol{\Sigma}}_{\beta_1} + \omega_{\beta_2} \sigma^2_{\beta_2} \tilde{\boldsymbol{\Sigma}}_{\beta_2}$$

where $\tilde{\boldsymbol{\Sigma}}_{\beta_i} = \boldsymbol{X}_i^T \boldsymbol{X}$ for $i = 1, 2$, and the rest is as we saw in Section 4.4.

We can proceed equivalently to enable selection between any partition of covariates, either applying the same procedure to $\boldsymbol{X}_1$ or $\boldsymbol{X}_2$ or both.

$$\omega_\beta \tilde{\boldsymbol{\Sigma}}_\beta = \sum_{i=1}^{S_\beta} \omega_\beta \sigma^2_{\beta_i} \tilde{\boldsymbol{\Sigma}}_{\beta_i} \tag{4.3}$$

and if we set priors with equal marginal variance on S' of the coefficient vectors and order them such that these come first, then their total covariance matrix is simply $\omega_{\beta_1} \sigma^2_{\beta_1} \sum_{i=1}^{S'} \tilde{\boldsymbol{\Sigma}}_{\beta_i}$, so the case in which we set the priors on a number of coefficient vectors to be the same, in other words the model in Equation (4.2), can be seen as a special case of Equation (4.3).

Chapter 5

Missing data examples

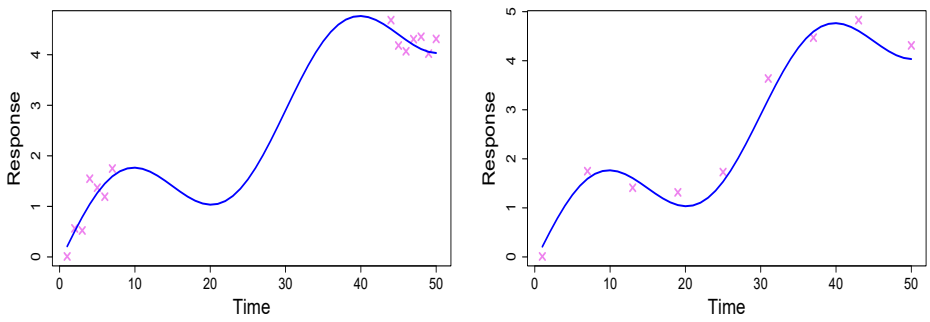
So far our model has been dealing with a very informative data set. As the data points cover the entire interval and are close together it is relatively easy to identify the residual error and by extension the non-linear effect. In more realistic settings, this is generally not guaranteed. It is thus natural to ask how our priors perform given more sparse information, and whether the ensuing difficulties in distinguishing between residual and non-linear effect variance would make it more appropriate to specify our priors differently, for instance in a way that seeks to coerce more strictly linear behaviour.

In this section, we aim to answer these questions by comparing model performance given the independent and HD priors from Chapter 3 and the expanded HD prior from Chapter 4 over two different cases of reduced data sets and two different prior specifications.

5.1 Reduced data sets and priors

Our two reduced data sets are chosen to pose two distinct problems. With the first we try to to investigate how well the model extrapolates over a single, long interval of unknown data points, and with the second we try to find how well it performs when the data points are still evenly spaced, but fewer in number.

More specifically, for the first data set, we chose to leave out 36 data points in one continuous interval centred on the middle of the data set, $n = 25$. In the latter we leave out 42 data points by including only every seventh data point. This has the desirable result that for $N = 50$ data points we have perfectly even spacing between every known data point whilst also including the end points, ensuring that every interval of unknown data points is adjacent to a known data point in either direction. Otherwise we would have the choice between ending



(a) First case (b) Second case

Figure 5.1: A visualisation of the reduced data sets used in Chapter 5. Again the blue curve denotes the true signal, whilst the known data points are displayed as violet crosses.

with an unknown data point, adding some extra "extrapolation error" which is besides the point for this problem, or putting two known data points more closely together, which would make the posterior of σ_R^2 easier to narrow down, diminishing the actual problem we are trying to pose. The ensuing data sets are illustrated in Figure 5.1.

From inspection of Figure 5.1 it appears these data sets should make inference substantially more difficult, as either could plausibly have been generated from a purely linear distribution, making it not immediately clear which choice of prior or prior belief would be best. We therefore test the prior specifications used in chapters 3 and 4, this time with a much less vague prior on σ_β for the former, against alternate specifications designed to coerce the model into behaving linearly.

Again, for the independent and basic HD priors, we choose to specify the priors such that median$(V') = 1$ and median$(\omega'_{\boldsymbol{u}}) = 0.25$. This time the prior on β is the less vague $\beta \sim \mathcal{N}(0, 6)$. We use primes to distinguish these parameters from those with the same symbol in the expanded HD prior. For the expanded HC prior we have chosen to encode the prior knowledge that the leaf node variances, $V_{\boldsymbol{\epsilon}}$, $V_{\boldsymbol{u}}$ and V_β are the same. In other words we choose median(V), median(ω_{S1}) and median$(\omega_{\boldsymbol{u}})$ such that $V_{\boldsymbol{u}} = \text{median}(\omega_{S1})\text{median}(\omega_{\boldsymbol{u}}) = 0.25$, similar to the specification for the independent and basic HD priors. The specification of the other parameters follows similarly.

The alternative prior specifications keep the same values for V', σ_β and V_β,

and set $\omega'_{\boldsymbol{u}} = 0.01$, $V_{\boldsymbol{\epsilon}} = 0.99$ and $V_{\boldsymbol{u}} = 0.01$.

For convenience, these new parameters are collected in Table 5.1.

Table 5.1: Prior specifications used in Chapter 5. Primes denote parameters for the independent and basic HD priors to avoid ambiguity, and V_A denotes the prior variance assigned to leaf node A.

Expert knowledge	Independent and HD prior $(V', \omega'_{\boldsymbol{u}}, \sigma^2_\beta)$	Expanded HD $(V_{\boldsymbol{\epsilon}}, V_{\boldsymbol{u}}, V_\beta)$
Base case	$(1, 0.25, 6)$	$(0.75, 0.25, 6)$
Linear model	$(1, 0.01, 6)$	$(0.99, 0.01, 6)$

5.2 Results

Our results are somewhat mixed regarding which prior is best. The CRPS and MSE scores over unknown data points in the first sparse data set are contained in Table 5.2, and the ensuing estimates, along with 95% credibility intervals are displayed in Figure 5.2. Here it appears that independent priors are the best in both cases, whilst the expanded HD prior is the worst by a significant margin. Both independent and basic HD priors change as expected given linearity coercing prior knowledge, producing tighter credibility intervals and slightly better scores, unlike the expanded HD prior. The HD prior for some reason does not becomes "more linear" given the altered prior knowledge, in stead it becomes simply performs worse, with a *less* rounded curve and worse scores.

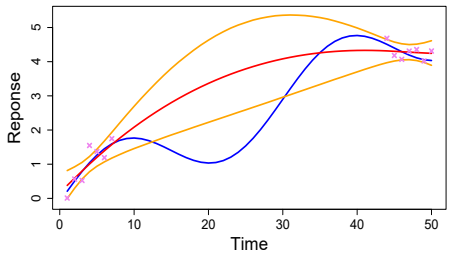
This is more or less the opposite of what we observe for the second data set, the results of which are compiled in Table 5.3 and illustrated in Figure 5.3. Although the priors behave similarly, this now leads to the expanded HD prior performing the best, in both prior knowledge cases, and the independent priors the worst. The expanded HD prior improves slightly in the case where we try to coerce linearity. From inspecting Figure 5.3 this appears to be for the same reasons that it performed the worst for the first reduced data set. Unlike the independent and basic HD prior, the expanded HR prior has not produced predominantly linear behaviour. This appears to be an advantage for this data set, as the model still manages to decently identify the non-linear effect. Ironically, it appears attempting to coerce linearity now leads the expanded HD prior to overfit. Had the data been noisier, this might have been a disadvantage, but the scores suggest it is not, at least for this data set. Interestingly it also appears the independent and basic HD priors perform more or less on par when linearity is coerced.

Prior knowledge	Prior type	Hold-out MSE	Hold-out CRPS
Basic	Independent	0.75	0.67
	HD	1.98	0.90
	HD+	2.95	0.98
Linear	Independent	0.71	0.70
	HD	0.81	0.67
	HD+	5.00	1.28

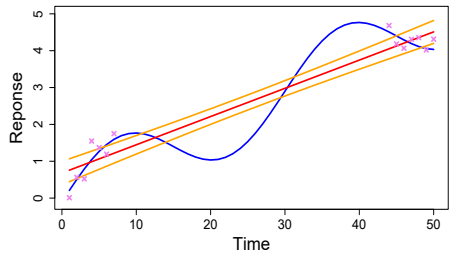
Table 5.2: Performance scores for the different priors given the first data set of Chapter 5.

Prior knowledge	Prior type	Hold-out MSE	Hold-out CRPS
Basic	Independent	0.54	0.54
	HD	0.40	0.42
	HD+	0.17	0.23
Linear	Independent	0.55	0.53
	HD	0.55	0.52
	HD+	0.13	0.20

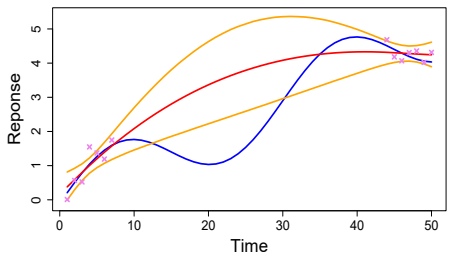
Table 5.3: Performance scores for the different priors given the second data set of Chapter 5.



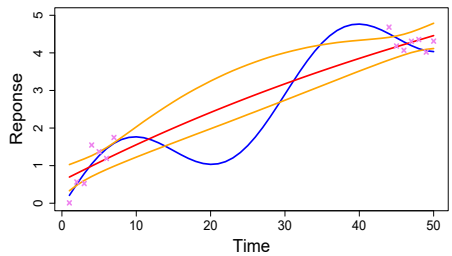
(a) Independent priors - Old prior knowledge



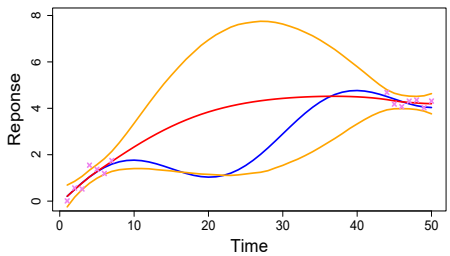
(b) Independent priors - Linear prior knowledge



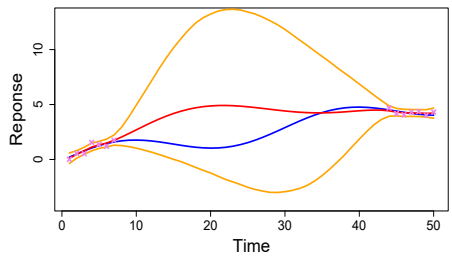
(c) HD prior - Old prior knowledge



(d) HD prior - Linear prior knowledge

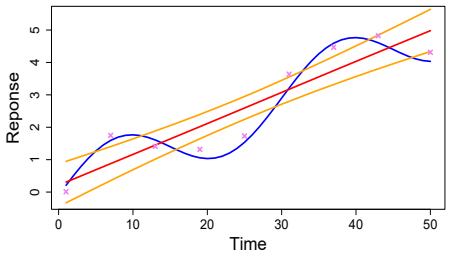


(e) Expanded HD prior - Old prior knowledge

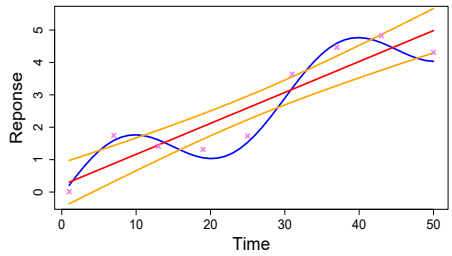


(f) Expanded HD prior - Linear prior knowledge

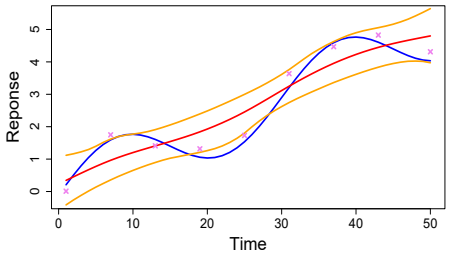
Figure 5.2: A comparison of model fits on the first reduced data set of Chapter 5. Blue curves display the true signal, red curves represent model estimates, orange curves show 95% credibility intervals, and known data points are plotted using violet crosses.



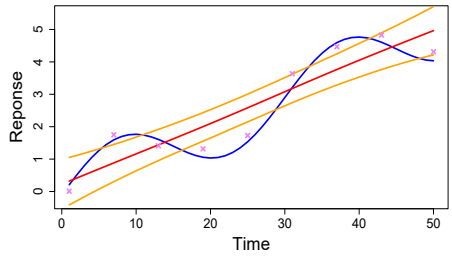
(a) Independent priors - Old prior knowledge



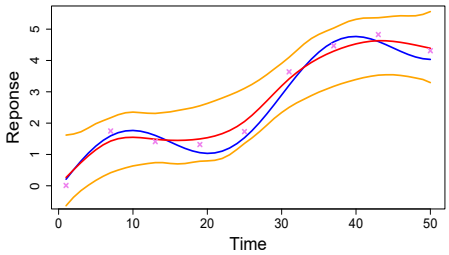
(b) Independent priors - Linear prior knowledge



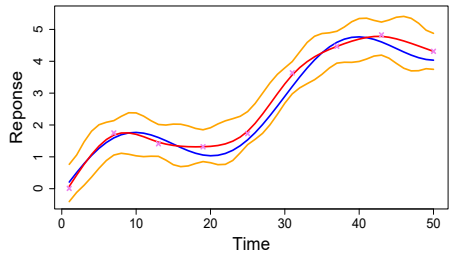
(c) HD prior - Linear prior knowledge



(d) HD prior - Linear prior knowledge



(e) Expanded HD prior - old prior knowledge



(f) Expanded HD prior - Old prior knowledge

Figure 5.3: A comparison of model fits for our second reduced data set of Chapter 5. Blue curves display the true signal, red curves represent model estimates, orange curves show 95% credibility intervals, and known data points are plotted using violet crosses.

Chapter 6

Multiple covariates - A simulation study

In Section 4.3, we briefly touched on the case of a model with multiple covariates. In this section we will be executing a simulation study on how the different types of priors considered so far perform in this new context. More specifically, we will be expanding the 1-dimensional smoothing problem from before to two dimensions, comparing prior performance by considering the ensuing scores on signal estimates as well as that of the contributions specific to either covariate. The priors in question again include independent and basic HD priors, as well as two expanded HD priors with differing tree structures.

6.1 Data and model likelihood

Our data will be generated similarly to that of Chapter 3. For now we will avoid any interaction between variables, so the likelihood is of the form

$$y_i \sim \mathcal{N}(\alpha + \sum_{j=1}^{p} f_j(x_{ji}), \sigma_R^2)\,,\ i = 1, 2, ..., n \tag{6.1}$$

where the total contribution f_i from the ith covariate x_i, $i = 1, 2$, is decomposed into a linear and non-linear component. We represent this, both when generating the data and in our model, using linear coefficients β_1 and β_2, and random-walk-2 vectors $\boldsymbol{u}$ and $\boldsymbol{w}$ for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ respectively, so $f_1(\boldsymbol{x}_1) = \beta_1\boldsymbol{x}_1 + \boldsymbol{u}$ and $f_2(\boldsymbol{x}_2) = \beta_2\boldsymbol{x}_2 + \boldsymbol{w}$. We do not include any interaction between covariates. Furthermore, to keep Stan runtimes manageable, we reduce the upper bound on permissible covariate values down from 50 to 10.

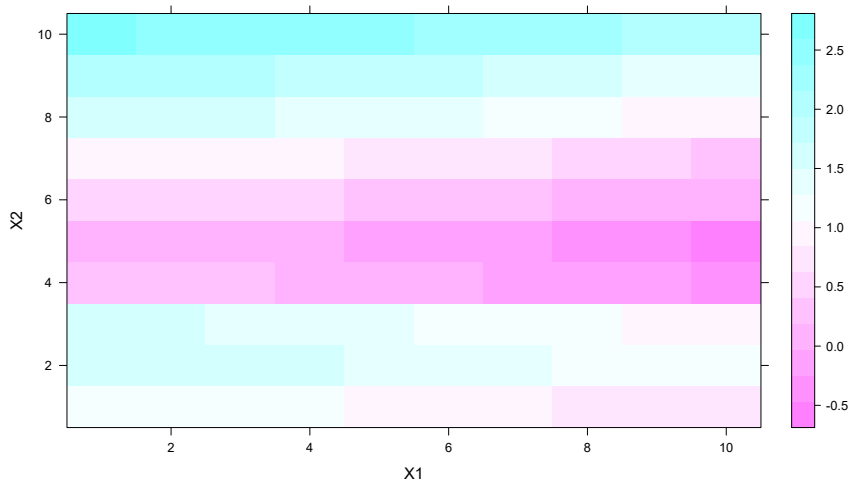


Figure 6.1: An example realisation of a signal generated as described in Chapter 6.

The distribution of the random walk components is analogous to that of Chapter 3. In other words

$$\Delta_2^2 u_i = u_{i+2} - 2u_{i+1} + u_i \sim \mathcal{N}(0, \sigma_{\boldsymbol{u}}^2)\,,\ i = 1, 2, ..., N-2 \tag{6.2}$$

$$\bar{\boldsymbol{u}} = \frac{1}{N}\sum_{n=1}^{N} u_n = 0 \quad \text{and} \quad \frac{\sum_{n=1}^{N} u_n (n - \frac{N+1}{2})}{\sum_{n=1}^{N} (n - \frac{N+1}{2})^2} = 0 \tag{6.3}$$

and equivalently for $\boldsymbol{w}$. Samples are obtained by generating standardised realisations of $\boldsymbol{u}$ and $\boldsymbol{w}$ and then scaling them by the desired standard deviations to produce samples. When generating data for the simulation study, we chose the parameter values $\alpha = 1$, $\boldsymbol{\beta} = (-0.2, -.3)$, $\sigma_{\boldsymbol{u}} = 0.05$ and $\sigma_{\boldsymbol{w}} = 1$. The signal of one such data set is displayed in Figure 6.1.

Moving parallel to the x_1 axis, one can see the predominantly linear behaviour of $f_1(\boldsymbol{x}_1)$, whereas moving vertically we can clearly see the noticeable non-linear behaviour of f_2, with a significant dip for x_1-values 4 to 6.

Figure 6.2: The HD prior "tree" for the independent prior for modelling a signal over two covariates. $\boldsymbol{u}$ and $\boldsymbol{w}$ are random walk components for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

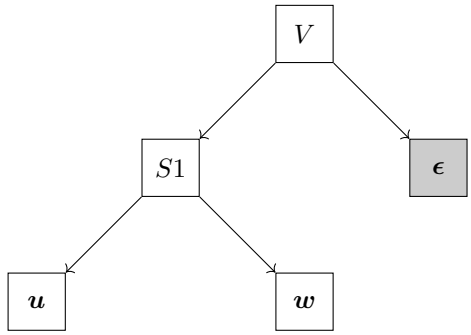


Figure 6.3: The HD prior tree the basic HD prior for two covariates. $\boldsymbol{u}$ and $\boldsymbol{w}$ are random walk components for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

6.2 Tree structures

Adding a covariate naturally necessitates differently structured and generally more complex priors. The changes to the independent prior(s) are the most basic, simply requiring a third isolated node, as illustrated in Figure 6.2. For this prior, as well as all others in this chapter, we encoded the prior belief that $V = 2.5$ (by making the median variances sum to the desired V and exhibit the desired proportions in this case, equivalently to the approach in Chapter 3).

The tree structures for the HD priors are more interesting. In all three cases we choose a PC prior on total variance, with a priori knowledge differing between basic HD priors and expanded ones similarly to before. For the most basic HD prior we enforce shrinkage towards residual variance, like in Section 3, with a PC prior split at the top of the tree with a 75% portion of total variance being assigned to $\boldsymbol{\epsilon}$ a priori. The next and only other split is then between $\boldsymbol{u}$ and $\boldsymbol{w}$. We see no reason to prefer one random walk effect over the other in the general case, and so we use an ignorance prior for this split. The ensuing structure is displayed in figure 6.3.

The greatest complexity thus far naturally comes from including also linear coefficients into the HD hierarchy. This complexity arguably also gives us a non-trivial choice regarding what tree structure to employ. A conventional tree

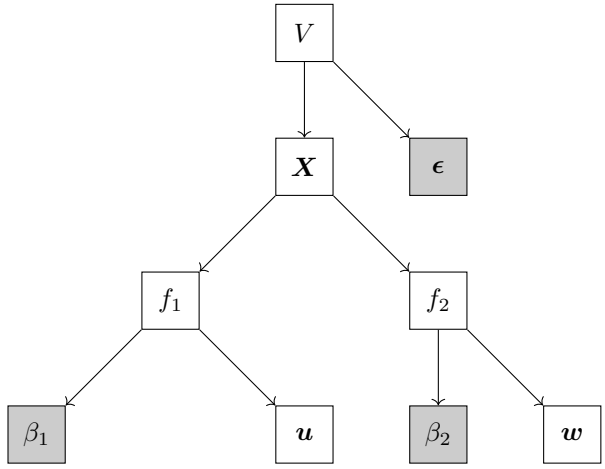


Figure 6.4: The expanded HD tree for two covariates with a Simpson type structure. $\boldsymbol{u}$ and $\boldsymbol{w}$ are random walk components in the same dimension as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

structure, and indeed one which has been used for a similar problem by Simpson et al. (2017), is the one displayed in Figure 6.4. Here we again start by enforcing shrinkage towards random noise using a PC split prior with 75% of total variance being assigned to $\boldsymbol{\epsilon}$ a priori. When choosing between $f_1(\boldsymbol{x}_1)$ and $f_2(\boldsymbol{x}_2)$ we again have no preference, and encode this using an ignorance split prior. For the remaining two splits we enforce shrinkage towards linearity, using PC split priors with 75% of total covariate function variance assigned to β_1 and β_2 a priori. We denote this expanded HD prior the Simpson HD+ prior.

We have also considered another tree structure using the same leaf nodes. Similar to how the HD priors in Chapter 3 were arguably parsimonious in how they enforce shrinkage, first preferring random noise over function variance, then linearity over non-linearity, this parsimonious prior starts with a PC prior assigning 75% of variance to $\boldsymbol{\epsilon}$. It then encodes equivalent shrinkage towards $\boldsymbol{\beta}$ in the split between $\boldsymbol{\beta}$ and $\boldsymbol{u}+\boldsymbol{w}$. Finally, it uses ignorance priors between the remaining leaf nodes. The ensuing structure is illustrated in Figure 6.5. We denote this the parsimonious HD+ prior.

To summarise, we have compiled the defining characteristics of each of the four priors in tables 6.1 and 6.2.

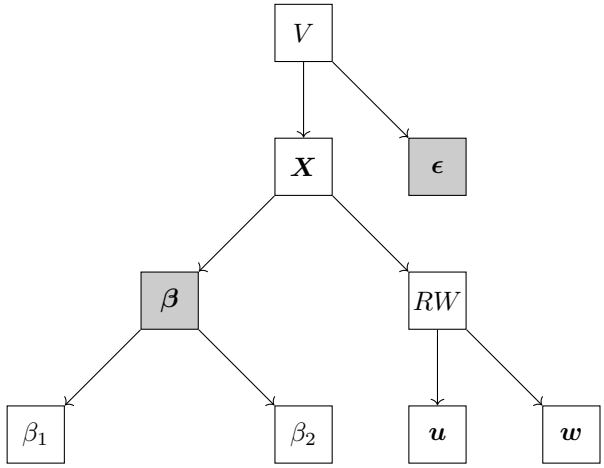


Figure 6.5: The expanded HD tree for two covariates with a "parsimonious" structure. $\boldsymbol{u}$ and $\boldsymbol{w}$ are random walk components for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

Prior	Total variance
Independent	2.5
Basic HD	2.5
Simpson	2.5
Parsimonious	2.5

Table 6.1: Total variances for the priors in Chapter 6.

Prior	Split	Prior knowledge
Independent[1]	$\boldsymbol{u}$, $\boldsymbol{w}$, $\boldsymbol{\epsilon}$	0.125, 0.125, 0.75
Basic HD	$\boldsymbol{u}+\boldsymbol{w}$, $\boldsymbol{\epsilon}$	0.25, 0.75
	$\boldsymbol{u}$, $\boldsymbol{w}$	Ignorant
Simpson	t, $\boldsymbol{\epsilon}$	0.25, 0.75
	f_1, f_2	Ignorant
	β_1, $\boldsymbol{u}$	0.75, 0.25
	β_2, $\boldsymbol{w}$	0.75, 0.25
Parsimonious	t, $\boldsymbol{\epsilon}$	0.25, 0.75
	$\boldsymbol{\beta}$, $\boldsymbol{u}+\boldsymbol{w}$	0.75, 0.25
	β_1, β_2	Ignorant
	$\boldsymbol{u}$, $\boldsymbol{w}$	Ignorant

Table 6.2: A summary of the priors in Chapter 6. A "split" of the form A, B means the split is between components A and B. "Prior knowledge" enumerates respective portions of total parent node variance assigned a priori, if any. Splits are ordered according to layer, starting at the root. t is the total latent model variance.

6.3 Simulation study and results

For our simulation study we generated 10 data sets with parameters $\alpha = 1$, $\boldsymbol{\beta} = (-0.2, 0.3)$, $\sigma_{\boldsymbol{u}} = \sqrt{0.05}$ and $\sigma_{\boldsymbol{w}} = 1$, holding out all but 10 randomly chosen points for each data set, and trained models on each data set using each of the four priors. The resulting scores for the signal estimate, as well as the estimates of the covariate specific contributions, $f_1(\boldsymbol{x}_1)$ and $f_2(\boldsymbol{x}_2)$ are compiled in Table 6.3.

Prior	Signal		$f_1(\boldsymbol{x}_1)$		$f_2(\boldsymbol{x}_2)$	
	MSE	CRPS	MSE	CRPS	MSE	CRPS
Independent	2.74	0.95	0.092	0.14	2.13	0.92
HD	0.39	0.35	0.056	0.12	0.19	0.24
Parsimonious HD	0.37	0.37	0.041	0.12	2.00	1.00
Simpson HD	0.58	0.49	0.057	0.15	2.02	1.00

Table 6.3: The average results from the simulation study in Chapter 6.

[1] We denote the portions of total variance assigned a priori to the different components by the independent prior even though it formally does not have any variance splits. This is similar to how the independent prior in Chapter 3 is described.

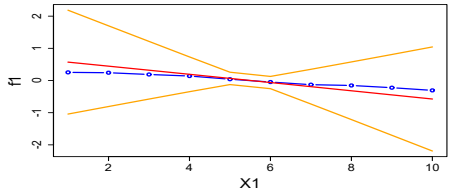
At first, some of these data might seem implausible. The errors for f_2 are relatively large compared to that of the signal overall for the expanded HD priors. However, there is an explanation for this, and it reiterates some of our results from Section 5.2. Recall that we hold out all but 10 randomly chosen points for each data set, and that we previously found that given a sparse data set, a model may perform better when reverting to one that is (almost) exclusively linear. As we can see from Figure 6.6, for every prior expect the basic HD prior, this has happened for both f_1 and f_2, although the ensuing incrase in error is only noticeable for f_2, as this is the only covariate function with a significant non-linear effect. Note that although we only show plots for one, this overall behaviour occurs for each data set.

To further support our explanation, we considered the number of effectively unique data points with respect to f_1 and f_2. Recall that f_1 and f_2 depend only on their respective covariate, so for every pair of data points that share a value of x_1 or x_2, the effective amount of information about that covariate's contribution is reduced. Naively, one might expect each covariate to have close to 10 unique values in most data sets, but we find that this is not the case. As can be seen in Table 6.4, most data sets have significantly less, particularly for x_2, with an average of 6.7 unique values per data set, and most having less than 8.

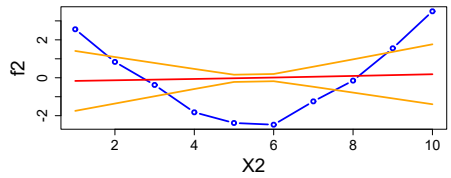
Data set	x_1	x_2
1	7	7
2	9	7
3	7	6
4	9	7
5	7	8
6	7	6
7	8	7
8	7	6
9	7	7
10	7	6
Mean	7.5	6.7

Table 6.4: Unique values of either covariate per data set used in Section 6.3.

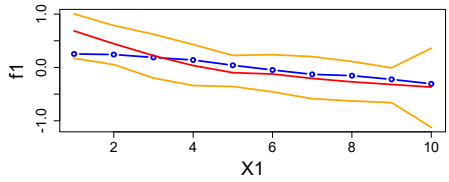
The other main insight from Table 6.3 is that the basic HD prior performs the best overall, having managed decent estimates of both covariate functions. This is a curious result given that this type of prior did not perform the best in any of our other examples so far, and it might be a subject of further research why only this prior could identify the non-linear effects, and more generally what situations cause some priors to fail or succeed at doing so.



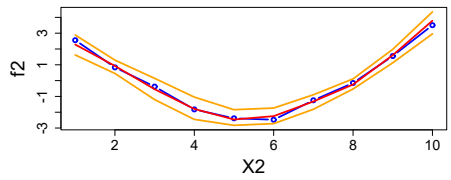
(a) Independent priors - f_1



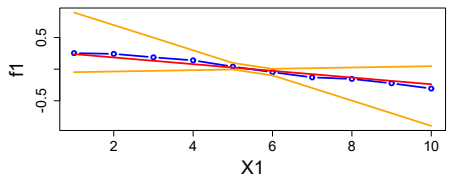
(b) Independent priors - f_2



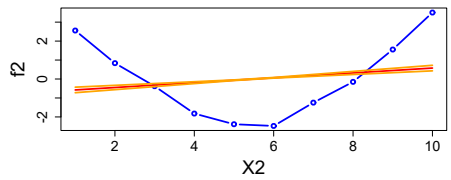
(c) Basic HD prior - f_1



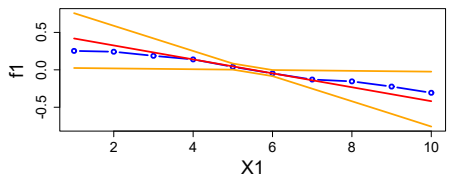
(d) Basic HD prior - f_2



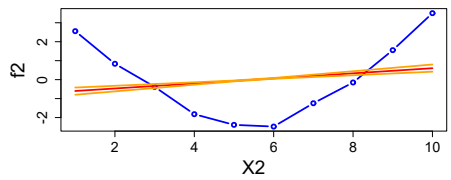
(e) Simpson HD+ prior - f_1



(f) Simpson HD+ prior - f_2



(g) Parsimonious HD+ prior - f_1



(h) Parsimonious HD+ prior - f_2

Figure 6.6: Example plots of true values (blue curve), estimates(red) and credibility intervals(orange) of f_1 (left column) and f_2 (right column) given the different priors.

Finally as a practical note, we stress that there are stark differences in how easily these priors can be fitted using Stan. As touched on in Section 2.4.5, performing decent Stan based inference might require adjustments to the model, like reparameterising in oter to decouple parameters and ensure their domains are without hard boundaries, but for difficult models this alone is not enough. In that case it is necessary to tune Stan's runtime arguments. For our inference we tried to maintain a mean effective samples size (ESS) of 5000 or more, with less than 1% divergent transitions, and preferably less than 10% of iterations exceeding the maximum tree depth for the integration stage of the algorithm, see Section 2.4.3. Sample ESS naturally increases with more iterations per chain, and the maximum tree depth, as incrementing this parameter doubles the potential range of each iteration. Incrementing maximum tree depth is naturally also the only way to reduce the number of transitions exceeding the maximum tree depth. Finally, to get rid of divergent transitions, we increase the target mean proposal acceptance probability during warmup. This forces the algorithm to take smaller steps when integrating, and thus explore the parameter space more slowly. However, adjusting any of these, particularly the latter two, come at the cost of substantially increased runtime per chain, so it is important to do so in moderation.

This had not been a major problem for our previous examples, but proved substantially worse here. To systematically approach this issue, we made use of a tuning function that starts by running a short test chain, checking for transitions that are divergent or exceed the maximum tree depth, and adjust target mean proposal acceptance probability and the maximum tree depth accordingly. If the portions of divergent and tree-depth-exceeding transitions fall beneath a given threshold, the function then adjust the number of iterations to achieve an acceptable mean ESS, and the process repeats, if needed.

When running this function on the model with independent priors, no increases to neither maximum tree depth nor target mean acceptance probability were needed, and the ensuing models were the fastest, completing in a matter of minutes. For the basic HD priors, some increments were needed, and runtimes were longer, taking hours, over half a day for some data sets. For the expanded HD priors, however, the chains could not be tuned to conform to our initial goals. After a few iterations, even the short test chains would take infeasibly long to run. We tried working around this by fixing the maximum tree depth at its default value and simply compensating with more iterations, but even after this there were still warnings about low ESS for some data sets. This might be another contributor to the low performance we have observed from these priors here. As is, these expanded priors seem too unstable to be feasibly used in this situation, though this might be due to the increased number of HD splits and not the expanded HR priors per se.

Chapter 7

Discussion

One of the key principal aims of PC and HD priors, as noted by both Simpson et al. (2017) and Fuglstad et al. (2020), is that the priors should be meaningful and intuitive. In general, through our work in this paper, we find this to be the case. In particular, even with the increase in complexity from the addition of another covariate in Chapter 6, the matter of specifying a prior was relatively straightforward, although there is a choice between first splitting variance between covariates, yielding the Simpson HD prior, or first performing shrinkage towards all linear effects, and then splitting between covariates, yielding the parsimonious HD prior.

Concerning the viability of the HD prior framework, our results are mixed. As seen in chapters 3 and 4, in the context of the 1-dimensional smoothing problem, the various HD priors perform on par with alternative priors given a complete data set, and possibly slightly better under mis-specification, which is most probably tied to the way HD priors encode prior knowledge more "softly", such that relative proportions of variance between components are left with more room to vary from the prior belief.

When comparing prior performance given incomplete data sets in Chapter 5, our results were mixed. For the first data set, with known data points concentreated at the end of the domain, the independent priors won out by a significant margin, and the expanded HD prior performed the worst. This failure might sow doubt regarding the overall viability of this new prior, as it failed to exhibit one of the key desired properties of PC, and by extension HD priors: Shrinkage towards the base model. More specifically, it did not shrink towards a predominantly linear model. This same property, however, appears to be what made it perform the best on the second data set. For that data set, known data points appeared at fixed intervals, and the independent prior performed the worst. Further research

may be needed to ascertain why the HD+ prior exhibits this property, and to get a general indication of when it will be desirable or not.

The simulation study of Section 6 provided more discouraging results. Here, the best prior overall was the basic HD prior, with every other prior failing to identify the non-linear effects in the data. The study also demonstrated that there is serious difficulty associated with performing Stan inference using the expanded HD priors. We are not sure about the degree to which this is due to the new priors themselves, or simply the increased tree complexity. Investigating this, as well as ways to alleviate these issues may be a subject of further research, but based on our current findings we cannot recommend using these priors in this context.

More research is is also needed into more complex models, as all the problems and priors considered were still relatively simple. Because there were no interaction between covariates, the problem in Chapter 6 was in a sense two 1-dimensional problems put together. The drastic increase in inference difficulty may also have negative implications for the use of expanded HD priors on more realistically complex problems with even more covariates. A simple compromise might be to simply revert to the basic HD prior framework, but even then the researcher might find the cost of increased complexity catching up with them, which is a problem the HD prior framework would have to overcome, should it become established as a viable "default" framework for joint variance priors in Bayesian hierarchical models. Alternatively, a workaround like working with a maximal tree height might be an answer, with a number of distinct HD trees for groups of components that per se do not lead to too complicated trees.

Bibliography

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.

Fuglstad, G.-A., Hem, I. G., Knight, A., Rue, H., and Riebler, A. (2020). Intuitive Joint Priors for Variance Parameters. *Bayesian Analysis*, 15(4):1109 – 1137.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515 – 534.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrák, M. (2020). Bayesian workflow. https://arxiv.org/abs/2011.01808.

Givens, G. H. and Hoeting, J. A. (2012). *Markov Chain Monte Carlo*, chapter 7, pages 201–235. John Wiley and Sons, Ltd.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359 – 378.

Hem, I. G., Fuglstad, G.-A., and Riebler, A. (2021). makemyprior: Intuitive construction of joint priors for variance parameters in r. https://arxiv.org/abs/2105.09712.

Hoffman, M. D. and Gelman, A. (2011). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. https://arxiv.org/abs/1111.4246.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields Theory and applications*. Number 1 in Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, Taylor and Francis Group, 6000 Broken Sound Parkway NW.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392.

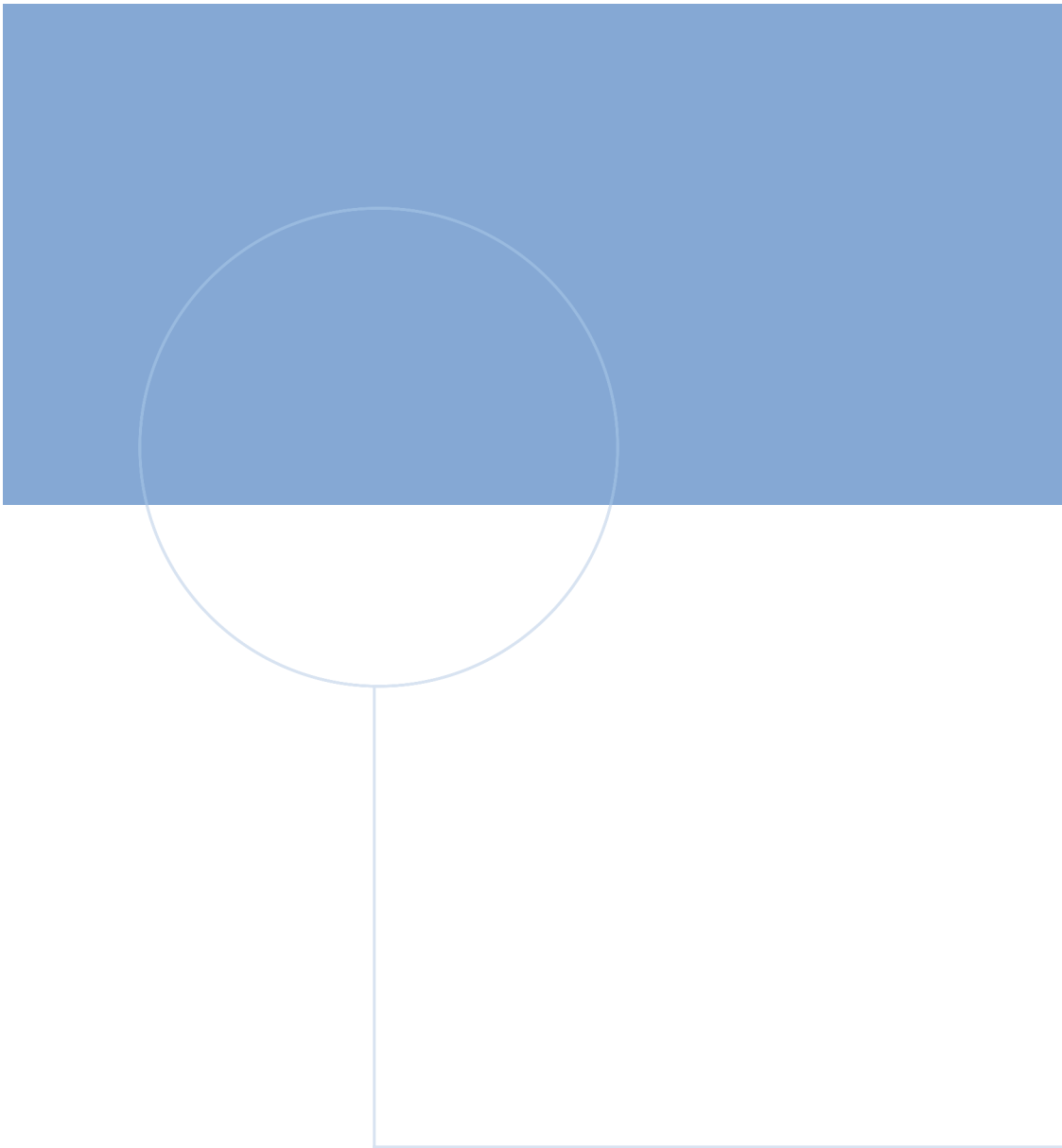
Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. (2016). Bayesian computing with inla: A review. https://arxiv.org/abs/1604.00860.

Simpson, D., Rue, H., Riebler, A., Martins, T. G., and Sørbye, S. H. (2017). Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors. *Statistical Science*, 32(1):1 – 28.

Spiegelhalter, D., Thomas, A., Best, N., and Gilks, W. (2000). Bugs 0.5* bayesian inference using gibbs sampling manual (version ii). *MRC Biostatistics unit, Institute of public health.*

Sørbye, S. H. and Rue, H. (2014). Scaling intrinsic gaussian markov random field priors in spatial modelling. *Spatial Statistics*, 8:39–51. Spatial Statistics Miami.

Yanchenko, E., Bondell, H. D., and Reich, B. J. (2021). The r2d2 prior for generalized linear mixed models. https://arxiv.org/abs/2111.10718.



NTNU

Kunnskap for en bedre verden