Deterministic Objective Bayesian Analysis for Spatial Models

Ryan Burn
ryan.burn@gmail.com

Abstract

Berger et al. (2001) and Ren et al. (2012) derived noninformative priors for Gaussian process models of spatially correlated data using the reference prior approach (Berger, Bernardo, 1991). The priors have good statistical properties and provide a basis for objective Bayesian analysis (Berger, 2006). Using a trust-region algorithm for optimization with exact equations for posterior derivatives and an adaptive sparse grid at Chebyshev nodes, this paper develops deterministic algorithms for fully Bayesian prediction and inference with the priors. Implementations of the algorithms are available at https://github.com/rnburn/bbai.

1 Introduction

Suppose we observe a Gaussian process $Z(\cdot)$ at sample points $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$ where

$$\begin{aligned} \mathbb{E}\left\{Z(\boldsymbol{s})\right\} &= \beta_1 x_1(\boldsymbol{s}) + \cdots + \beta_p x_p(\boldsymbol{s}) \\ &= \boldsymbol{\beta}' \boldsymbol{x}(\boldsymbol{s}); \\ \operatorname{cov}\left\{Z(\boldsymbol{s}), Z(\mathbf{u})\right\} &= \sigma^2 \left\{\psi_\ell(\|\boldsymbol{s} - \mathbf{u}\|) + \eta\right\}; \end{aligned} \tag{1}$$

$\boldsymbol{x}(\cdot)$ and $\psi_\ell(\cdot)$ represent the known regressor function and correlation function; and $\boldsymbol{\beta}$, σ^2, ℓ, and η represent the unknown regression coefficients, signal variance, length, and noise-to-signal ratio.

Let $\boldsymbol{\theta}$ denote the unknown parameters $\left(\boldsymbol{\beta}, \sigma^2, \ell, \eta\right)'$. To reason about possible values at unobserved points, we'd like to know the distribution $\mathrm{P}\left(Z(\mathbf{u}) \mid \mathbf{y}, \boldsymbol{\theta}_{\text{true}}\right)$ where $\mathbf{u}$ is an unobserved point and $\mathbf{y}$ denotes the observations $\left(Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_n)\right)'$. Of course, different values of $\boldsymbol{\theta}$ could reasonably produce $\mathbf{y}$, so there's no way we can identify $\boldsymbol{\theta}_{\text{true}}$ or construct prediction distributions exactly. We need ways to approximate.

Approach 1: Maximize Likelihood

Suppose the likelihood function, $L(\boldsymbol{\theta}; \mathbf{y}) \propto \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\theta}\right)$, is strongly peaked about an optimum, $\boldsymbol{\theta}_{\text{ml}}$. Then $\boldsymbol{\theta}_{\text{true}}$ should be close to $\boldsymbol{\theta}_{\text{ml}}$, and $\mathrm{P}\left(Z(\mathbf{u}) \mid \mathbf{y}, \boldsymbol{\theta}_{\text{ml}}\right)$ should be a reasonable substitute for $\mathrm{P}\left(Z(\mathbf{u}) \mid \mathbf{y}, \boldsymbol{\theta}_{\text{true}}\right)$.

i	s	y	i	s	y
1	0.00	6.34	11	0.53	2.25
2	0.05	1.62	12	0.58	4.30
3	0.11	7.38	13	0.63	-4.40
4	0.16	12.22	14	0.68	-2.54
5	0.21	3.03	15	0.74	10.94
6	0.26	-4.58	16	0.79	-2.81
7	0.32	-3.45	17	0.84	-2.82
8	0.37	-4.48	18	0.89	2.53
9	0.42	-8.02	19	0.95	10.01
10	0.47	2.61	20	1.00	1.52

Table 1: Randomly sampled data from Gaussian process (2)

.

But what happens if a broad range of parameters could reasonably produce $\mathbf{y}$?

Example 1.1. [source] Consider the data set from Table 1. I randomly sampled the Gaussian process (1) with

$$\sigma^2 = 25, \quad \ell = 0.01, \quad \eta = 0.1, \text{ and } \psi_\ell(t) = \exp\left\{-\frac{t^2}{2\ell^2}\right\} \tag{2}$$

at 20 evenly spaced points on the interval $[0, 1]$. Likelihood has a maximum at

$$\sigma^2_{\text{ml}} = 34.42, \quad \ell_{\text{ml}} = 0.035, \quad \text{and } \eta_{\text{ml}} = 3.82 \times 10^{-6}.$$

Note how much smaller η_{ml} is than its true value. If we try to use $\boldsymbol{\theta}_{\text{ml}}$ as a substitute for $\boldsymbol{\theta}_{\text{true}}$, we will get bad results as Figure 1 shows. Put

$$g(t) = L(\boldsymbol{\theta}_{\text{ml}}(1-t) + \boldsymbol{\theta}_{\text{true}}t; \mathbf{y})/L(\boldsymbol{\theta}_{\text{ml}}; \mathbf{y}).$$

$g(\cdot)$ computes the relative likelihood along a line segment from $\boldsymbol{\theta}_{\text{ml}}$ to $\boldsymbol{\theta}_{\text{true}}$, and Figure 2 plots $g(t)$ for $0 \leq t \leq 1$. Looking at the figure, we can confirm that likelihood is not strongly peaked about an optimum and any value of $\boldsymbol{\theta}$ along the line segment could have reasonably produced $\mathbf{y}$.

Approach 2: Integrate over Possible Parameters

We saw in Example 1.1 that using maximum likelihood parameters can lead to poor results when the likelihood function isn't strongly peaked (Berger et al., 1999). Instead of approximating prediction distributions with only a single value

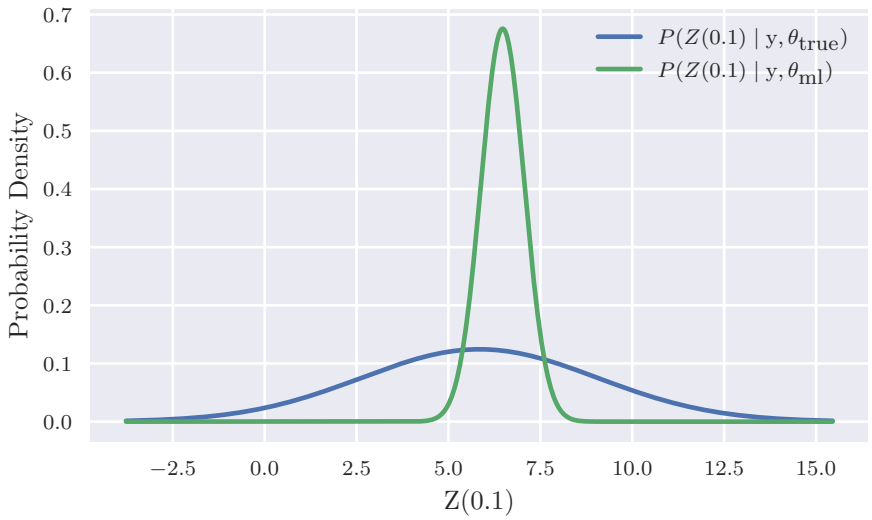

Figure 1: Compare Gaussian process prediction distributions $\mathrm{P}\left(Z(0.1) \mid \mathbf{y}, \boldsymbol{\theta}_{\mathrm{ml}}\right)$ and $\mathrm{P}\left(Z(0.1) \mid \mathbf{y}, \boldsymbol{\theta}_{\mathrm{true}}\right)$

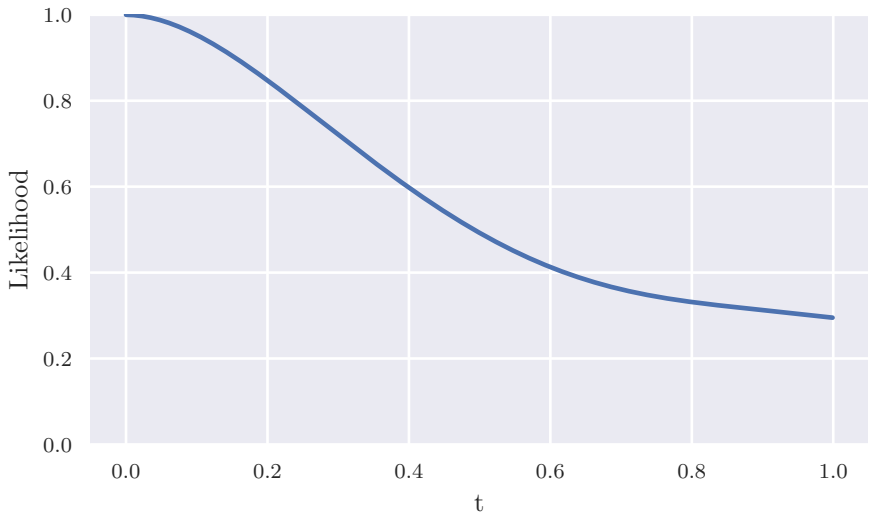

Figure 2: Relative likelihood for different values of $\boldsymbol{\theta}$ on the line segment from $\boldsymbol{\theta}_{\mathrm{ml}}$ to $\boldsymbol{\theta}_{\mathrm{true}}$

of $\boldsymbol{\theta}$, let's consider every $\boldsymbol{\theta}$ and weigh by a posterior distribution, $\pi(\boldsymbol{\theta} \mid \mathbf{y})$,

$$\mathrm{P}^{\pi}(Z(\mathbf{u}) \mid \mathbf{y}) = \int \mathrm{P}(Z(\mathbf{u}) \mid \mathbf{y}, \boldsymbol{\theta})\, \pi(\boldsymbol{\theta} \mid \mathbf{y})\, d\boldsymbol{\theta}.$$

$\pi(\boldsymbol{\theta} \mid \mathbf{y})$ measures our belief that parameters $\boldsymbol{\theta}$ generated the observations $\mathbf{y}$. To derive $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, we apply Bayes' theorem: $\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto L(\boldsymbol{\theta}; \mathbf{y}) \times \pi(\boldsymbol{\theta})$ where $\pi(\boldsymbol{\theta})$ measures our prior belief that the model has parameters $\boldsymbol{\theta}$.

Naturally, this leads to the question: How do we specify $\pi(\boldsymbol{\theta})$ when we know nothing particular about $\boldsymbol{\theta}$? Statisticians have grappled with the problem of specifying so-called noninformative priors ever since Bayes and Laplace first started applying the approach to the binomial model over 200 years ago.

While noninformative priors continue to be debated, fortunately, the modern approach of reference priors gives a general path forward and, particularly, for the case of Gaussian processes works quite well.

Before getting into the details (see §3 and §4 for descriptions of the prior and prediction algorithm), let's look at how the approach works on the Gaussian process from Example 1.1.

> **Example 1.2.** [source] (Example 1.1 continued) In Figure 3, I plot the prediction distribution for the Example 1.1 data set using the Bayesian approach with a reference prior and compare to the true prediction distribution. We can see that the Bayesian approach gives a better approximation to the true prediction distribution than the maximum likelihood approach, Figure 1.

2 How to Specify Noninformative Priors

The goal of a noninformative prior is to represent "minimal information" so that inference is driven by the data and the model rather than prior knowledge.

Making this goal exact is difficult; and it's unlikely there will ever be a universal approach to noninformative priors that's optimal for all situations, as there can be multiple reasonable definitions of "minimal information". However, frequentist coverage has emerged as one key metric to test whether a candidate noninformative prior is suitable for objective Bayesian analysis. Here's the basic idea: Let $\Theta_1 \times \cdots \times \Theta_k$ denote the parameter space for the model, pick α to be something like 0.95, and run Algorithm 1 for different $\tilde{\boldsymbol{\theta}}$ varied across the model's parameter space. If the prior is good, Algorithm 1 should produce a result close to α.

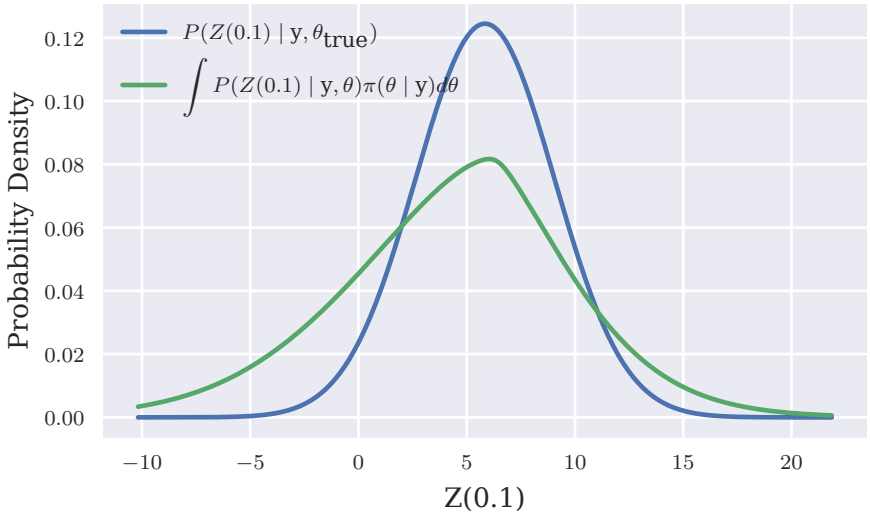

Figure 3: Compare the prediction distribution from the Bayesian approach with reference prior, $\mathrm{P}^{\pi}\left(Z(0.1) \mid \mathbf{y}\right)$, to the true prediction distribution $\mathrm{P}\left(Z(0.1) \mid \mathbf{y}, \boldsymbol{\theta}_{\text{true}}\right)$.

Algorithm 1 Test accuracy of credible sets produced with a prior

function coverage-test($\tilde{\boldsymbol{\theta}}$, j, α)
  $cnt \leftarrow 0$
  $N \leftarrow$ a large number
  for $i \leftarrow 1$ to N **do**
    $\tilde{\mathbf{y}} \leftarrow$ sample from $P(\cdot \mid \tilde{\boldsymbol{\theta}})$
    $\tilde{\boldsymbol{\Theta}} \leftarrow \Theta_1 \times \cdots \times \Theta_{j-1} \times \Theta_j \cap (-\infty, \tilde{\theta}_j] \times \Theta_{j+1} \times \cdots \times \Theta_k$
    $t \leftarrow \int_{\tilde{\boldsymbol{\Theta}}} \pi(\boldsymbol{\theta} \mid \tilde{\mathbf{y}}) d\boldsymbol{\theta}$
    if $\frac{\alpha}{2} < t < 1 - \frac{\alpha}{2}$ **then**
      $cnt \leftarrow cnt + 1$
    end if
  end for
  return $\frac{cnt}{N}$
end function

With Algorithm 1 in our toolbox, let's look at a few approaches for specifying noninformative priors.

Constant Prior

We begin with the simplest approach: Set $\pi\left(\boldsymbol{\theta}\right) \propto 1$. Immediately, we see one serious disadvantage of this approach: It's not invariant under reparameteri-

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
σ^2	coverage	σ^2	coverage	σ^2	coverage	σ^2	coverage
0.1	0.9502	0.1	0.9486	0.1	0.9508	0.1	0.9493
0.5	0.9519	0.5	0.9478	0.5	0.9492	0.5	0.9488
1.0	0.9516	1.0	0.9495	1.0	0.9517	1.0	0.9494
2.0	0.9514	2.0	0.9521	2.0	0.9539	2.0	0.9489
5.0	0.9489	5.0	0.9455	5.0	0.9558	5.0	0.9488

Table 2: Frequentist coverages for the mean of a normal distribution with known variance and constant prior.

zation. If $\varphi(\cdot)$ is some strictly increasing function onto $[a, b]$ with continuous derivative, then

$$\int_a^b L(\theta; \mathbf{y}) d\theta = \int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} L(\varphi(u); \mathbf{y}) \dot{\varphi}(u) du.$$

Thus, different parameterizations with the constant prior lead to different posterior distributions.

Still, let's try the approach out on some examples.

Example 2.1. [source] Suppose we observe n normally distributed values, $\mathbf{y}$, with variance 1 and unknown mean, μ. Then

$$\begin{aligned} L(\mu; \mathbf{y}) &\propto \exp\left\{-\frac{1}{2}(\mathbf{y} - \mu\mathbf{1})'(\mathbf{y} - \mu\mathbf{1})\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(n\mu^2 - 2\mu n\bar{y}\right)\right\} \\ &\propto \exp\left\{-\frac{n}{2}(\mu - \bar{y})^2\right\}. \end{aligned}$$

Thus,

$$\int_{-\infty}^{t} \pi(\mu \mid \mathbf{y})\, d\mu = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{t - \bar{y}}{\sqrt{2/n}}\right)\right].$$

I ran Algorithm 1 for $N = 10{,}000$, $\alpha = 0.95$, and various values of μ and n. Table 2 shows the results.

Example 2.2. [source] Suppose we observe n normally distributed values, $\mathbf{y}$, with zero-mean and unknown variance, σ^2. Then $L(\sigma^2; \mathbf{y}) \propto$

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
σ^2	coverage	σ^2	coverage	σ^2	coverage	σ^2	coverage
0.1	0.9014	0.1	0.9288	0.1	0.9418	0.1	0.9439
0.5	0.9035	0.5	0.9309	0.5	0.9415	0.5	0.9398
1.0	0.9048	1.0	0.9303	1.0	0.9404	1.0	0.9412
2.0	0.9079	2.0	0.9331	2.0	0.9402	2.0	0.9393
5.0	0.9023	5.0	0.9295	5.0	0.9339	5.0	0.9426

Table 3: Frequentist coverages for the variance of a normal distribution with known mean and constant prior.

$\left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{ns^2}{2\sigma^2}\right\}$ where $s^2 = \frac{\mathbf{y}'\mathbf{y}}{n}$. Put $u = \frac{ns^2}{2\sigma^2}$. Then

$$\int_0^t \left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{ns^2}{2\sigma^2}\right\} d\sigma^2 \propto \int_{\frac{ns^2}{2t}}^{\infty} u^{n/2-2} \exp\{-u\}\, du$$
$$= \Gamma(\frac{n-2}{2}, \frac{ns^2}{2t}).$$

Thus,

$$\int_0^t \pi\left(\sigma^2 \mid \mathbf{y}\right) d\sigma^2 = \frac{1}{\Gamma(\frac{n-2}{2})}\Gamma(\frac{n-2}{2}, \frac{ns^2}{2t}).$$

I ran Algorithm 1 for $N = 10{,}000$, $\alpha = 0.95$, and various values of σ^2 and n. Table 3 shows the results.

In Example 2.1, the constant prior produces nearly perfect results. In Example 2.2, the prior is notably off for smaller values of n but improves as n increases.

Jeffreys Prior

Dissatisfied with the inconsistency of the constant prior under reparameterization, Harold Jeffreys searched for a better approach and proposed the prior $\pi(\boldsymbol{\theta}) \propto |\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})|^{1/2}$ where $\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})$ is the Fisher information matrix,

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})_{st} = \mathbb{E}_{\mathbf{y}}\left\{\left(\frac{\partial}{\partial\theta_s}\log \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\theta}\right)\right)\left(\frac{\partial}{\partial\theta_t}\log \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\theta}\right)\right)\right\}.$$

We can check that unlike the constant prior, Jeffreys prior is invariant to reparameterization: If $\boldsymbol{\varphi}(\mathbf{u})$ is an injective continuously differentiable function whose range includes Θ and whose Jacobian is never zero on $\boldsymbol{\varphi}^{-1}(\Theta)$, then the change

of variables formula gives us

$$\int_\Theta L(\boldsymbol{\theta};\mathbf{y})|\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})|^{1/2}d\boldsymbol{\theta} = \int_{\boldsymbol{\varphi}^{-1}(\Theta)} L(\boldsymbol{\varphi}(\mathbf{u});\mathbf{y})|\boldsymbol{\mathcal{I}}(\boldsymbol{\varphi}(\mathbf{u}))|^{1/2} \mid (|\boldsymbol{D\varphi}(\mathbf{u})|) \mid d\mathbf{u},$$

where $\boldsymbol{D\varphi}(\mathbf{u})$ denotes the Jacobian matrix $\boldsymbol{D\varphi}(\mathbf{u})_{st} = \frac{\partial\varphi_s(\mathbf{u})}{\partial u_t}$. Let $\boldsymbol{\mathcal{I}}^{\boldsymbol{\varphi}}(\mathbf{u})$ denote the Fisher information matrix with respect to the reparameterization. Then

$$\begin{aligned}
\boldsymbol{\mathcal{I}}^{\boldsymbol{\varphi}}(\mathbf{u})_{st} &= \mathbb{E}_{\mathbf{y}}\left\{\left(\frac{\partial}{\partial u_s}\log \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\varphi}(\mathbf{u})\right)\right)\left(\frac{\partial}{\partial u_t}\log \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\varphi}(\mathbf{u})\right)\right)\right\} \\
&= \mathbb{E}_{\mathbf{y}}\left\{\left(\nabla_{\boldsymbol{\theta}}\log \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\theta}\right)'\frac{\partial\boldsymbol{\varphi}}{\partial u_s}(\mathbf{u})\right)\left(\nabla_{\boldsymbol{\theta}}\log \mathrm{P}\left(\mathbf{y} \mid \boldsymbol{\theta}\right)'\frac{\partial\boldsymbol{\varphi}}{\partial u_t}(\mathbf{u})\right)\right\} \\
&= \left(\frac{\partial\boldsymbol{\varphi}}{\partial u_s}(\mathbf{u})\right)'\boldsymbol{\mathcal{I}}(\varphi(\mathbf{u}))\left(\frac{\partial\boldsymbol{\varphi}}{\partial u_t}(\mathbf{u})\right).
\end{aligned}$$

Thus, $\boldsymbol{\mathcal{I}}^{\boldsymbol{\varphi}}(\mathbf{u}) = \boldsymbol{D\varphi}(\mathbf{u})'\boldsymbol{\mathcal{I}}(\boldsymbol{\varphi}(\mathbf{u}))\boldsymbol{D\varphi}(\mathbf{u})$ and

$$\int_\Theta L(\boldsymbol{\theta};\mathbf{y})|\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})|^{1/2}d\boldsymbol{\theta} = \int_{\boldsymbol{\varphi}^{-1}(\Theta)} L(\boldsymbol{\varphi}(\mathbf{u});\mathbf{y})|\boldsymbol{\mathcal{I}}^{\boldsymbol{\varphi}}(\mathbf{u}))|^{1/2}d\mathbf{u}.$$

Example 2.3. (Example 2.1 continued) To compute the Fisher information matrix, we first differentiate $\log L(\mu;\mathbf{y})$,

$$\begin{aligned}
\frac{\partial}{\partial\mu}\log L(\mu;\mathbf{y}) &= \frac{\partial}{\partial\mu}\left(-\frac{n}{2}\left(\mu-\bar{y}\right)^2\right) \\
&= -n\left(\mu-\bar{y}\right).
\end{aligned}$$

Then we compute $\mathbb{E}_{\mathbf{y}}\left\{\left(\frac{\partial}{\partial\mu}L(\mu;\mathbf{y})\right)^2 \mid \mu\right\} = \mathbb{E}_{\mathbf{y}}\left\{n^2\left(\mu-\bar{y}\right)^2 \mid \mu\right\}$. $\bar{y}-\mu$ is normally distributed with zero mean and variance $\frac{1}{n}$, so $\mathbb{E}_{\mathbf{y}}\left\{\left(\frac{\partial}{\partial\mu}L(\mu;\mathbf{y})\right)^2 \mid \mu\right\} = n$. Jeffreys prior in this case is the same as the constant prior.

Example 2.4. [source] (Example 2.2 continued) We differentiate $\log L(\sigma^2;\mathbf{y})$ to get

$$\begin{aligned}
\frac{\partial}{\partial\sigma^2}\log L(\sigma^2;\mathbf{y}) &= \frac{\partial}{\partial\sigma^2}\left(-\frac{n}{2}\log\sigma^2 - \frac{ns^2}{2\sigma^2}\right) \\
&= \frac{n}{2\sigma^2}\left(\frac{s^2}{\sigma^2}-1\right).
\end{aligned}$$

Now, $\mathbb{E}_{\mathbf{y}}\left\{\left(\frac{\partial}{\partial\sigma^2}L(\sigma^2;\mathbf{y})\right)^2 \mid \sigma^2\right\} = \left(\frac{n}{2\sigma^2}\right)^2\mathbb{E}_{\mathbf{y}}\left\{\left(\frac{s^2}{\sigma^2}-1\right)^2 \mid \sigma^2\right\}$ and $y_1^2 + \cdots + y_n^2$ follows a chi-squared distribution and with variance $2n\sigma^4$ and

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
σ^2	coverage	σ^2	coverage	σ^2	coverage	σ^2	coverage
0.1	0.9516	0.1	0.9503	0.1	0.9509	0.1	0.9511
0.5	0.9501	0.5	0.949	0.5	0.952	0.5	0.948
1.0	0.9505	1.0	0.9511	1.0	0.9513	1.0	0.95
2.0	0.948	2.0	0.9514	2.0	0.9501	2.0	0.9482
5.0	0.9506	5.0	0.9497	5.0	0.9486	5.0	0.9485

Table 4: Frequentist coverages for the variance of a normal distribution with known mean and Jeffreys prior.

mean $n\sigma^2$, so

$$\begin{aligned}\mathbb{E}_{\mathbf{y}}\left\{s^4 \mid \sigma^2\right\} &= \frac{\sigma^4\left(2n+n^2\right)}{n^2} \\ &= \sigma^4\left(1+\frac{2}{n}\right)\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}_{\mathbf{y}}\left\{\left(\frac{\partial}{\partial\sigma^2}L(\sigma^2;\mathbf{y})\right)^2 \mid \sigma^2\right\} &= \left(\frac{n}{2\sigma^2}\right)^2\mathbb{E}_{\mathbf{y}}\left\{\frac{s^4}{\sigma^4}-2\frac{s^2}{\sigma^2}+1 \mid \sigma^2\right\} \\ &= \left(\frac{n}{2\sigma^2}\right)^2\left(\frac{2}{n}\right) \\ &= \frac{n}{2\sigma^4}.\end{aligned}$$

We derive the prior $\pi\left(\sigma^2\right) \propto \frac{1}{\sigma^2}$. For the CDF, we apply the same derivations in Example 2.2 to get

$$\int_0^t \pi\left(\sigma^2 \mid \mathbf{y}\right) d\sigma^2 = \frac{1}{\Gamma(\frac{n}{2})}\Gamma(\frac{n}{2}, \frac{ns^2}{2t}).$$

Using the same setup in Example 2.2, I produced the coverages in Table 4.

So far, Jeffreys prior performs excellently. In fact, for a single parameter, Welch, Peers (1963) show that in the limiting case, coverage for $(1-\alpha)\,\%$ credible sets using Jeffreys prior approaches α with an asymptotic error $o(n^{-1})$. Moreover, it's the only prior with this property, so starting with the goal of matching coverage naturally leads us to Jeffreys prior.

Let's check how well Jeffeys prior performs in cases with more than a single variable.

Example 2.5. [source] Suppose we observe n normally distributed values,

$\mathbf{y}$, with unknown mean, μ, and unknown variance, σ^2. Then

$$L(\mu, \sigma^2; \mathbf{y}) \propto \left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mu\mathbf{1})'(\mathbf{y} - \mu\mathbf{1})\right\}.$$

We differentiate $\log L(\cdot; \mathbf{y})$ to get

$$\frac{\partial}{\partial\mu}\log L(\mu, \sigma^2; \mathbf{y}) = \frac{n}{\sigma^2}(\bar{y} - \mu) \quad \text{and}$$
$$\frac{\partial}{\partial\sigma^2}\log L(\mu, \sigma^2; \mathbf{y}) = -\frac{n}{2}\frac{1}{\sigma^2} + \frac{1}{2}\left(\frac{1}{\sigma^2}\right)^2 (\mathbf{y} - \mu\mathbf{1})'(\mathbf{y} - \mu\mathbf{1}).$$

We apply the derivations from Example 2.3 and Example 2.4 to get the Fisher information matrix $\boldsymbol{\mathcal{I}}(\mu, \sigma^2) = \begin{pmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{pmatrix}$ and the Jeffreys prior $\pi\left(\mu, \sigma^2\right) \propto \left(\frac{1}{\sigma^2}\right)^{3/2}$. Let's check coverage for σ^2. First, we integrate out μ,

$$\begin{aligned}
&\int_{-\infty}^{\infty} L(\mu, \sigma^2; \mathbf{y})\, \pi\left(\mu, \sigma^2\right) d\mu \\
&\qquad \propto \int_{-\infty}^{\infty} \left(\frac{1}{\sigma^2}\right)^{(n+3)/2} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{y} - \mu\mathbf{1}\|^2\right\} d\mu \\
&\qquad = \left(\frac{1}{\sigma^2}\right)^{(n+3)/2} \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)\right\} \\
&\qquad\qquad \int_{-\infty}^{\infty} \exp\left\{-\frac{n}{2\sigma^2}(\mu - \bar{y})^2\right\} d\mu \\
&\qquad \propto \left(\frac{1}{\sigma^2}\right)^{(n+2)/2} \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)\right\}.
\end{aligned}$$

Then

$$\int_0^t \int_{-\infty}^{\infty} \pi\left(\mu, \sigma^2 \mid \mathbf{y}\right) d\mu d\sigma^2 = \frac{1}{\Gamma(\frac{n}{2})}\Gamma(\frac{n}{2}, \frac{1}{2t}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)).$$

I ran Algorithm 1 for $N = 10{,}000$, $\alpha = 0.95$, $\mu = 0$, and various values of σ^2 to get the results in Table 5.

Unfortunately, the multiparameter case is not so easy; and as we see in Example 2.5, Jeffreys prior doesn't perform nearly as well. Jeffreys considered modifications of his prior to handle the multiparameter case better but never developed a rigorous approach. For that, we turn to reference priors.

$n=5$		$n=10$		$n=15$		$n=20$	
σ^2	coverage	σ^2	coverage	σ^2	coverage	σ^2	coverage
0.1	0.9241	0.1	0.938	0.1	0.9419	0.1	0.9463
0.5	0.9219	0.5	0.9377	0.5	0.946	0.5	0.9441
1.0	0.9245	1.0	0.94	1.0	0.9431	1.0	0.944
2.0	0.9236	2.0	0.9391	2.0	0.9446	2.0	0.9432
5.0	0.9182	5.0	0.9395	5.0	0.9403	5.0	0.9458

Table 5: Frequentist coverages for the variance of a normal distribution with unknown mean and Jeffreys prior.

Reference Priors

If Jeffreys prior works well in the single parameter case, why not apply it to parameters one at a time? In the reference prior approach (Berger, Bernardo, 1991), we build up a multiparameter prior by marginalizing the likelihood with a conditional prior of fewer parameters to form a new integrated likelihood function with only a single parameter, to which we can apply Jeffreys prior.

Suppose $L(\theta_1, \theta_2; \mathbf{y})$ is a likelihood function of two variables. We fix θ_1 and use Jeffreys' approach to derive a conditional prior $\pi(\theta_2 \mid \theta_1)$. Then we integrate out θ_2,

$$L^I(\theta_1; \mathbf{y}) = \int_{\Theta_2} L(\theta_1, \theta_2; \mathbf{y})\, \pi(\theta_2 \mid \theta_1)\, d\theta_2,$$

to get the integrated likelihood function $L^I(\cdot; \mathbf{y})$ of only a single variable. We apply Jefferys' approach again to the integrated likelihood function to get $\pi(\theta_1)$ and form the complete prior $\pi(\theta_1, \theta_2) = \pi(\theta_1) \times \pi(\theta_2 \mid \theta_1)$. If the prior $\pi(\cdot \mid \theta_1)$ is improper, we can choose a sequence of compact subsets $A_1 \subset A_2 \subset \cdots \subset \Theta_2$ such that $\lim_{t\to\infty} A_t = \Theta_2$, apply the approach to A_t, and take the limit as $t \to \infty$.

Let's try this out on Example 2.5.

Example 2.6. [source] (Example 2.5 continued). We first integrate out μ using the constant conditional prior,

$$\begin{aligned} L^I(\sigma^2; \mathbf{y}) &= \int_{-\infty}^{\infty} L(\mu, \sigma^2; \mathbf{y})\, \pi\left(\mu \mid \sigma^2\right) d\mu \\ &\propto \left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{n}{\sigma^2}\left(\mu - \bar{y}\right)^2\right\} d\mu \\ &\propto \left(\frac{1}{\sigma^2}\right)^{(n-1)/2} \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)\right\}. \end{aligned}$$

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
σ^2	coverage	σ^2	coverage	σ^2	coverage	σ^2	coverage
0.1	0.9533	0.1	0.948	0.1	0.9504	0.1	0.9519
0.5	0.9528	0.5	0.9499	0.5	0.9524	0.5	0.9486
1.0	0.948	1.0	0.9503	1.0	0.9507	1.0	0.9484
2.0	0.9529	2.0	0.9504	2.0	0.9515	2.0	0.9487
5.0	0.9525	5.0	0.9507	5.0	0.9484	5.0	0.9511

Table 6: Frequentist coverages for the variance of a normal distribution with unknown mean and reference prior.

Now, we differentiate $L^I(\cdot; \mathbf{y})$ to find the Fisher information matrix,

$$\begin{aligned}\frac{\partial}{\partial \sigma^2} \log L^I(\sigma^2; \mathbf{y}) &= -\frac{n-1}{2\sigma^2} + \frac{1}{2}\left(\frac{1}{\sigma^2}\right)^2 \left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right) \\ &= \frac{1}{2\sigma^2}\left\{\frac{1}{\sigma^2}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right) - (n-1)\right\}.\end{aligned}$$

Put $Z = \frac{1}{\sigma^2}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)$. Then Z follows a chi-squared distribution with $n-1$ degrees of freedom so that $\mathbb{E}[Z] = n-1$, $\mathbb{E}[Z^2] = 2(n-1) + (n-1)^2$, and

$$\begin{aligned}\boldsymbol{\mathcal{I}}(\sigma^2) &= \left(\frac{1}{2\sigma^2}\right)^2 \left\{\mathbb{E}[Z^2] - 2(n-1)\,\mathbb{E}[Z] + (n-1)^2\right\} \\ &= \frac{n-1}{2\sigma^4}.\end{aligned}$$

Thus, we derive the reference prior $\pi\left(\mu, \sigma^2\right) = \frac{1}{\sigma^2}$. Following Example 2.5, we compute

$$\int_0^t \int_{-\infty}^{\infty} \pi\left(\mu, \sigma^2 \mid \mathbf{y}\right) d\mu d\sigma^2 = \frac{1}{\Gamma(\frac{n-1}{2})}\Gamma(\frac{n-1}{2}, \frac{1}{2t}\left(\mathbf{y}'\mathbf{y} - n\bar{y}^2\right)).$$

I reran the coverage simulation from Example 2.5 with this CDF and got the results in Table 6. Comparing to Table 5, we can see that the reference prior approach gives better results.

3 Noninformative Priors for Spatial Models

Let's consider noninformative priors for the Gaussian process (1).

- Using a constant prior isn't a viable option. In addition to the problem of incoherence, the resulting posterior would be improper (Berger, 2006).

We might consider truncating the parameter space to make the constant prior proper, but that doesn't solve the problem as inference would be highly dependent on the truncation bounds.

- Certain modified forms of Jeffreys prior result in a proper posterior, but the credible sets produced from the priors perform poorly (Ren et al., 2012).

That brings us to the reference prior approach. Since the model has multiple parameters, we'll first integrate out $\boldsymbol{\beta}$ and σ^2 using the conditional prior $\pi\left(\boldsymbol{\beta}, \sigma^2 \mid \ell, \eta\right) \propto \frac{1}{\sigma^2}$. Likelihood for Gaussian process (1) is given by

$$L(\boldsymbol{\beta}, \sigma^2, \ell, \eta; \mathbf{y}) \propto \left(\sigma^2\right)^{-n/2} |\mathbf{G}|^{-1/2} \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)' \mathbf{G}^{-1}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right\}$$

where $\mathbf{X} = (\boldsymbol{x}(\boldsymbol{s}_1), \ldots, \boldsymbol{x}(\boldsymbol{s}_n))'$, $\mathbf{G} = \eta \boldsymbol{I} + \mathbf{K}(\ell)$, and $\mathbf{K}(\ell)_{ij} = \psi_\ell\left(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|\right)$. Integrating likelihood with the conditional prior gives us

$$\begin{aligned} L^I(\ell, \eta; \mathbf{y}) &\propto \int_0^\infty \int_{\mathbb{R}^p} L(\boldsymbol{\beta}, \sigma^2, \ell, \eta; \mathbf{y})\, \pi\left(\boldsymbol{\beta}, \sigma^2 \mid \ell, \eta\right) d\boldsymbol{\beta} d\sigma^2 \\ &\propto \int_0^\infty \left(\sigma^2\right)^{-(n-p)/2} |\mathbf{G}|^{-1/2} |\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}|^{-1/2} \exp\left\{-\frac{S^2}{2\sigma^2}\right\}\left(\frac{1}{\sigma^2}\right) d\sigma^2 \\ &\propto |\mathbf{G}|^{-1/2} |\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}|^{-1/2} \left(S^2\right)^{-(n-p)/2} \end{aligned} \tag{3}$$

where

$$S^2 = \mathbf{y}'\boldsymbol{R}\mathbf{y} \quad \text{and} \quad \boldsymbol{R} = \mathbf{G}^{-1} - \mathbf{G}^{-1}\mathbf{X}\left(\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{G}^{-1}. \tag{4}$$

After computing the Fisher information matrix for $L^I(\cdot; \mathbf{y})$ and forming its Jeffrey prior, we derive the complete prior

$$\pi\left(\boldsymbol{\beta}, \sigma^2, \ell, \eta\right) \propto \left(\frac{1}{\sigma^2}\right) |\boldsymbol{\Sigma}(\ell, \eta)|^{1/2} \tag{5}$$

where

$$\boldsymbol{\Sigma}(\ell, \eta) = \begin{pmatrix} \operatorname{tr}\left\{\left(\boldsymbol{R}\frac{\partial \mathbf{K}}{\partial \ell}\right)^2\right\} & \operatorname{tr}\left(\boldsymbol{R}^2 \frac{\partial \mathbf{K}}{\partial \ell}\right) & \operatorname{tr}\left(\boldsymbol{R}\frac{\partial \mathbf{K}}{\partial \ell}\right) \\ * & \operatorname{tr}(\boldsymbol{R}^2) & \operatorname{tr}(\boldsymbol{R}) \\ * & * & n-p \end{pmatrix}. \tag{6}$$

For a detailed derivation, see Ren et al. (2012).

To test the performance of the reference prior, we'll run the same simulations used in Ren et al. (2012). Details of how to compute the integrals will be given in §4.

> **Example 3.1.** [source] To generate observations, I sample Gaussian pro-

	$\eta = 0.01$			$\eta = 0.05$		
	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$
ℓ coverage	0.945	0.985	0.995	0.950	0.990	1.000
η coverage	0.885	0.980	0.995	1.000	0.995	1.000
σ^2 coverage	0.990	0.995	0.980	0.975	0.985	0.985
β_1 coverage	1.000	0.990	0.965	0.995	0.995	0.945
	$\eta = 0.1$			$\eta = 0.2$		
	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$
ℓ coverage	0.965	0.975	0.995	0.985	1.000	1.000
η coverage	1.000	0.975	0.995	0.995	0.970	0.990
σ^2 coverage	1.000	0.985	0.985	0.970	0.985	0.980
β_1 coverage	0.995	0.980	0.955	0.995	0.985	0.930

Table 7: Frequentist coverages for Gaussian process parameters on simulation data sets with a constant regressor.

cess (1) with

$$\sigma^2 = 1, \quad x_1(\boldsymbol{s}) = 1, \quad \boldsymbol{\beta}_1 = 1, \quad \text{and} \quad \psi_\ell(d) = \exp\left\{-\frac{d}{\ell}\right\}$$

at 10×10 evenly spaced points on the interval $[0, 1] \times [0, 1]$. I ran Algorithm 1 with $N = 200$ and allowed ℓ and η to vary. The results are given in Table 7.

Example 3.2 (source)**.** For the next simulation, I modify the Gaussian process in Example 3.1 to include additional regressors, $\boldsymbol{x}((u, v)) = \left(1, u, v, u^2, uv, v^2\right)'$ with $\boldsymbol{\beta} = (0.15, -0.65, -0.1, 0.9, -1.0, 1.2)'$. Rerunning the simulation experiment with the same values of ℓ and η gave the coverages in Table 8.

To test prediction performance, we can use a modified form of Algorithm 1.

	$\eta = 0.01$			$\eta = 0.05$		
	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$
ℓ coverage	0.995	1.000	0.960	1.000	1.000	0.910
η coverage	0.865	0.950	0.915	1.000	1.000	0.990
σ^2 coverage	0.995	0.975	0.835	1.000	0.985	0.760
β_1 coverage	1.000	0.925	0.765	0.960	0.915	0.775
β_2 coverage	0.945	0.895	0.870	0.935	0.900	0.845
β_3 coverage	0.990	0.885	0.850	0.960	0.915	0.895
β_4 coverage	0.915	0.900	0.835	0.940	0.890	0.855
β_5 coverage	0.970	0.860	0.890	0.970	0.935	0.870
β_6 coverage	0.935	0.900	0.840	0.955	0.910	0.875
	$\eta = 0.1$			$\eta = 0.2$		
	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$
ℓ coverage	1.000	1.000	0.890	1.000	1.000	0.815
η coverage	0.990	1.000	1.000	0.990	1.000	1.000
σ^2 coverage	0.990	0.980	0.835	0.985	0.975	0.835
β_1 coverage	0.980	0.870	0.795	0.960	0.900	0.800
β_2 coverage	0.925	0.910	0.895	0.965	0.925	0.850
β_3 coverage	0.970	0.915	0.865	0.940	0.900	0.905
β_4 coverage	0.940	0.920	0.900	0.940	0.915	0.885
β_5 coverage	0.945	0.870	0.895	0.960	0.865	0.870
β_6 coverage	0.965	0.885	0.860	0.940	0.925	0.940

Table 8: Frequentist coverages for Gaussian process parameters on simulation data sets with polynomial regressors.

	$\eta = 0.001$			$\eta = 0.01$		
	$\ell = 0.1$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 0.1$	$\ell = 0.2$	$\ell = 0.5$
Bay coverage	0.919	0.951	0.942	0.939	0.953	0.944
ML coverage	0.812	0.905	0.934	0.838	0.912	0.919
	$\eta = 0.1$			$\eta = 0.2$		
	$\ell = 0.1$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 0.1$	$\ell = 0.2$	$\ell = 0.5$
Bay coverage	0.929	0.943	0.932	0.936	0.937	0.938
ML coverage	0.847	0.893	0.920	0.853	0.893	0.903

Table 9: Frequentist coverages for Bayesian and maximum likelihood Gaussian process predictions on simulation data sets.

Algorithm 2 Test accuracy of prediction credible sets produced with a prior

function prediction-coverage-test($\tilde{\boldsymbol{\theta}}$, α)
  $cnt \leftarrow 0$
  $N \leftarrow$ a large number
  for $i \leftarrow 1$ to N **do**
    $\tilde{\mathbf{y}} \leftarrow$ sample from $P(\cdot \mid \tilde{\boldsymbol{\theta}})$
    $t \leftarrow \int_{-\infty}^{\tilde{y}_1} \int \mathrm{P}\left(y' \mid \boldsymbol{\theta}\right) \pi(\boldsymbol{\theta} \mid \tilde{y}_2, \ldots, \tilde{y}_n) d\boldsymbol{\theta} dy'$
    if $\frac{\alpha}{2} < t < 1 - \frac{\alpha}{2}$ **then**
      $cnt \leftarrow cnt + 1$
    end if
  end for
  return $\frac{cnt}{N}$
end function

Example 3.3. [source] To generate observations, I sample from Gaussian process (1) with $\sigma^2 = 1$ and $\psi_\ell(d) = \exp\left\{-\frac{d^2}{2\ell^2}\right\}$. I sampled training observations at 20 evenly spaced points on the interval $[0, 1]$ and test observations at random points on the interval $[0, 1]$. I ran Algorithm 2 with $N = 100$ and varied ℓ and η. Table 9 shows the coverage results for Bayesian prediction distributions using the reference prior and maximum likelihood prediction distributions.

4 Deterministic Bayesian Inference

The key component for deterministic prediction and inference is an accurate approximation to the posterior distribution for ℓ and η that enables efficient computation of integrals, $\tilde{\pi}\left(\ell, \eta \mid \mathbf{y}\right) \approx L^I(\ell, \eta; \mathbf{y}) \times \pi\left(\ell, \eta\right)$ where $\pi\left(\ell, \eta\right) \propto |\boldsymbol{\Sigma}(\ell, \eta)|^{1/2}$ and $\boldsymbol{\Sigma}(\cdot)$ is defined in (6).

Given $\tilde{\pi}(\cdot \mid \mathbf{y})$, it's relatively straightforward to derive approximations for the marginal distributions

$$\begin{aligned}
\pi(\ell \mid \mathbf{y}) &\approx \int_0^\infty \tilde{\pi}(\ell, \eta \mid \mathbf{y})\, d\eta, \\
\pi(\eta \mid \mathbf{y}) &\approx \int_0^\infty \tilde{\pi}(\ell, \eta \mid \mathbf{y})\, d\ell, \quad \text{and} \\
\pi\left(\sigma^2 \mid \mathbf{y}\right) &\approx \int_0^\infty \int_0^\infty \mathrm{P}^\pi\left(\sigma^2 \mid \mathbf{y}, \ell, \eta\right) \tilde{\pi}(\ell, \eta \mid \mathbf{y})\, d\ell d\eta
\end{aligned}$$

and approximations for prediction distributions,

$$\mathrm{P}^\pi(Z(\boldsymbol{s}) \mid \mathbf{y}) \approx \int_0^\infty \int_0^\infty \mathrm{P}^\pi(Z(\boldsymbol{s}) \mid \mathbf{y}, \ell, \eta)\, \tilde{\pi}(\ell, \eta \mid \mathbf{y})\, d\ell d\eta.$$

Outline of Algorithm

Assume $\varphi_\ell(\cdot)$ and $\varphi_\eta(\cdot)$ are strictly increasing functions onto $(0, \infty)$ with continuous derivatives. Put

$$\begin{aligned}
f(\mathbf{u}) = & -\log L^I(\varphi_\ell(u_1), \varphi_\eta(u_2); \mathbf{y}) \\
& -\log \pi(\varphi_\ell(u_1), \varphi_\eta(u_2)) - \log \dot{\varphi}_\ell(u_1) - \log \dot{\varphi}_\eta(u_2).
\end{aligned} \tag{7}$$

$f(\cdot)$ is the negative log of the reparameterized posterior $\pi(\ell, \eta \mid \mathbf{y})$. Approximation of $\exp(-f(\cdot))$ naturally leads to approximation and integration of $\pi(\ell, \eta \mid \mathbf{y})$.

We'll build an approximation in four steps.

Algorithm 3 Build a multivariate polynomial to approximate $\exp(-f(\cdot))$ where $f(\cdot)$ (7) is the negative negative log of the reparameterized posterior function $\pi(\ell, \eta \mid \mathbf{y})$

1: Using a trust-region optimizer and exact equations for ∇f and $\nabla^2 f$, minimize f to find $\mathbf{u}_{\text{map}}$.

2: Let $\mathbf{v}_1$ and $\mathbf{v}_2$ denote two orthonormal eigenvectors of the Hessian at $\mathbf{u}_{\text{map}}$, $\nabla^2 f(\mathbf{u}_{\text{map}})$. Find values $a_1 < 0 < b_1$ and $a_2 < 0 < b_2$ such that

$$\begin{aligned} -\left(f(\mathbf{u}_{\text{map}} + a_i \mathbf{v}_i) - f(\mathbf{u}_{\text{map}})\right) &= \log \varepsilon_1(a_i) \quad \text{and} \\ -\left(f(\mathbf{u}_{\text{map}} + b_i \mathbf{v}_i) - f(\mathbf{u}_{\text{map}})\right) &= \log \varepsilon_2(b_i) \end{aligned}$$

for $i = 1,\ 2$ and $\varepsilon_i(\cdot)$ small. These values bracket $f(\cdot)$ around a rectangular region oriented along the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ that contains most of the probability mass.

3: Find monotonic cubic splines $s_1(\cdot)$ and $s_2(\cdot)$ such that $s_i(0) = a_i$, $s_i(0.5) = 0$, and $s_i(1) = b_i$ for $i = 1, 2$.

4: Put

$$g(\boldsymbol{x}) = \exp\left\{-\left(f\left(\mathbf{u}_{\text{map}} + s_1(x_1)\mathbf{v}_1 + s_2(x_2)\mathbf{v}_2\right) - f(\mathbf{u}_{\text{map}})\right)\right\}. \tag{8}$$

Using Chebyshev nodes and the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ for a basis, adaptively build a sparse grid and interpolating polynomial to approximate $g(\cdot)$ (and hence $\pi(\ell, \eta \mid \mathbf{y})$) over the region $[0, 1] \times [0, 1]$.

Proposition 7 and Proposition 9 from Ren et al. (2012) show that $\pi(\ell, \eta \mid \mathbf{y})$ is bounded as $\ell \to 0$ or $\eta \to 0$ and derive $\mathcal{O}(\cdot)$ functions for when $\ell \to \infty$ and $\eta \to \infty$. Using suitable choices of $\varphi_\ell(\cdot)$, $\varphi_\eta(\cdot)$, and $\varepsilon_i(\cdot)$, we can achieve bounds for the probability mass outside of the bracketing region in Step 2. Following Gu et al. (2018), we use the parameterization $\phi_\ell(t) = \phi_\eta(t) = \exp(t)$. We'll only consider the simple case of ε_i fixed to some small constant, but other choices could lead to tighter bounding.

We can use any decent root-finding algorithm (e.g., Newton's method) for Step 2; we use the monotonic cubic algorithm from Fritsch, Carlson (1980) for Step 3. Step 1 and Step 4 are more complicated, and I break them down in greater detail in the next sections.

Step 1: Trust-region Optimization

Let $f\colon \mathbb{R}^p \to \mathbb{R}$ denote a twice-differentiable objective function. Trust-region methods are iterative, second-order optimization algorithms that produce a sequence $\{\boldsymbol{x}_k\}$ where the k^{th} iteration is generated by updating the previous

iteration with a solution to the subproblem (Sorensen, 1982)

$$\begin{aligned}\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + \boldsymbol{s}_k \quad \text{and} \\ \boldsymbol{s}_k &= \underset{\boldsymbol{s}}{\operatorname{argmin}} \left\{ \nabla f(\boldsymbol{x}_{k-1})'\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}'\nabla^2 f(\boldsymbol{x}_{k-1})\boldsymbol{s} \right\} \\ &\qquad \text{such that } \|\boldsymbol{s}\| \leq \delta_k.\end{aligned}$$

The subproblem minimizes the second-order approximation of f at $\boldsymbol{x}_{k-1}$ within the neighborhood $\|\boldsymbol{s}\| \leq \delta_k$, called the trust region. Using the trust region, we can restrict the second-order approximation to areas where it models f well. Efficient algorithms exist to solve the subproblem regardless of whether $\nabla^2 f(\boldsymbol{x}_{k-1})$ is positive-definite, making trust-region methods well-suited for non-convex optimization problems (Moré, Sorensen, 1983). With proper rules for updating δ_k and standard assumptions, such as Lipschitz continuity of ∇f, trust-region methods are globally convergent. Moreover, if $\nabla^2 f$ is Lipschitz continuous for all $\boldsymbol{x}$ sufficiently close to a nondegenerate second-order stationary point $\boldsymbol{x}_*$ where $\nabla^2 f(\boldsymbol{x}_*)$ is positive-definite, then trust-region methods have quadratic local convergence (Nocedal, Wright, 2006).

Algorithm 4 describes the trust-region algorithm we use for Step 1, and Appendix A derives equations for evaluating the value, gradient, and Hessian of the objective (7).

Algorithm 4 Minimize an objective function $f(\cdot)$

function minimize($f, \boldsymbol{x}_0$)
  $tol \leftarrow$ tolerance
  $\delta_0 \leftarrow$ an initial trust-region radius
  $y_0 \leftarrow f(\boldsymbol{x}_0)$
  $\boldsymbol{g}_0 \leftarrow \nabla f(\boldsymbol{x}_0)$
  $\boldsymbol{H}_0 \leftarrow \nabla^2 f(\boldsymbol{x}_0)$
  $k \leftarrow 0$
  while $\|\boldsymbol{g}_k\|_\infty > tol$ or $\boldsymbol{H}_k$ is not positive definite **do**
    $\boldsymbol{x}_{k+1}$, y_{k+1}, $\delta_{k+1} \leftarrow$ compute-next-step($\boldsymbol{x}_k$, y_k, $\boldsymbol{g}_k$, $\boldsymbol{H}_k$, δ_k)
    $\boldsymbol{g}_{k+1} \leftarrow \nabla f(\boldsymbol{x}_{k+1})$
    $\boldsymbol{H}_{k+1} \leftarrow \nabla^2 f(\boldsymbol{x}_{k+1})$
    $k \leftarrow k + 1$
  end while
  return $\boldsymbol{x}_k$, y_k, $\boldsymbol{H}_k$
end function
function compute-next-step($\boldsymbol{x}_k$, y_k, $\boldsymbol{g}_k$, $\boldsymbol{H}_k$, δ_k)
  $\delta_{k+1} \leftarrow \delta_k$
  while 1 **do**
    $\boldsymbol{s}_k \leftarrow \operatorname{argmin}_{\boldsymbol{s}} \left\{ \boldsymbol{g}_k' \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}' \boldsymbol{H}_k \boldsymbol{s} \;\middle|\; \|\boldsymbol{s}\| \le \delta_{k+1} \right\}$
      ▷ Solve the trust-region subproblem (Moré, Sorensen, 1983)
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{s}_k$
    $y_{k+1} \leftarrow f(\boldsymbol{x}_{k+1})$
    $\rho \leftarrow \frac{y_{k+1} - y_k}{\boldsymbol{g}_k' \boldsymbol{s}_k + \frac{1}{2} \boldsymbol{s}_k' \boldsymbol{H}_k \boldsymbol{s}_k}$
      ▷ ρ measures the accuracy of the second-order Taylor approximation to $f(\cdot)$ within the trust-region neighborhood, δ_{k+1}, about $\boldsymbol{x}_k$
    if $\rho < \frac{1}{4}$ **then**
      $\delta_{k+1} \leftarrow \frac{1}{4} \delta_{k+1}$ ▷ Shrink the trust region
    else if $\rho > \frac{3}{4}$ and $\|\boldsymbol{s}_k\| = \delta_{k+1}$ **then**
      $\delta_{k+1} \leftarrow 2\delta_{k+1}$ ▷ Expand the trust region
    end if
    if $\rho > \frac{1}{4}$ **then**
      return $\boldsymbol{x}_{k+1}$, y_{k+1}, δ_{k+1}
    end if
  end while
end function

Step 4: Sparse Grid Approximation

We seek to approximate $g(\cdot)$ (8) by a polynomial $\tilde{g}(\cdot)$ that interpolates $g(\cdot)$ at points in $[0, 1] \times [0, 1]$. If we choose the points well, we can achieve high accuracy with a minimal number of points, making $\tilde{g}(\cdot)$ cheaper to build and evaluate.

The simplest approach would be to interpolate at equispaced points, but polynomials at equispaced points perform terribly (see Runge's phenomenon). Much better is to interpolate at Chebyshev nodes. Polynomials at Chebyshev nodes have excellent approximation performance (Trefethen, 2019), but interpolating on a dense grid would still be expensive. We can achieve better efficiency if we interpolate on a sparse grid, and we can achieve even better efficiency if we adaptively construct the sparse grid to avoid unnecessary evaluations in areas that can be approximated well by lower-order polynomials.

Put

$$\begin{aligned} X^i &= \left\{x_1^i, \ldots, x_{m_i}^i\right\}, \\ m_i &= \begin{cases} 1 & \text{if } i = 0, \\ 2^{i-1} + 1 & \text{otherwise,} \end{cases} \\ x_j^i &= \begin{cases} \frac{1}{2} & \text{if } i = 0, \\ \frac{1}{2}\left(1 - \cos\frac{\pi(j-1)}{m_i - 1}\right) & \text{otherwise.} \end{cases} \end{aligned}$$

The Chebyshev-Gauss-Lobatto nodes, $\left\{X^i\right\}$, form a nested sequence of points, $X^i \subset X^{i+1}$, that serve as a building block for constructing interpolations and quadrature rules for sparse grids (Barthelmann et al., 2000; Klimke, 2006). Let $\psi_j^i(\cdot)$ denote the unique $(m_i - 1)$-degree polynomial where

$$\psi_j^i(x_{j'}^i) = \begin{cases} 1 & \text{if } j = j', \\ 0 & \text{otherwise;} \end{cases}$$

let V^i denote the vector space spanned by the basis functions $\left\{\psi_j^i\right\}$ for $j = 1, \ldots, m_i$; and define $\Delta V^0 = V^0$, $\Delta V^i = V^i - V^{i-1}$ for $i > 0$. We will build an approximation using functions from vector spaces

$$W^{\mathcal{I}} = \bigoplus_{\boldsymbol{i} \in \mathcal{I}} \Delta V^{i_1} \otimes \cdots \otimes \Delta V^{i_d}$$

where the index set $\mathcal{I}$ is required to be *admissible*: if $\boldsymbol{i} \in \mathcal{I}$ and $i_k > 0$, then $\boldsymbol{i} - \boldsymbol{e}_k \in \mathcal{I}$. The vector spaces $W^{\mathcal{I}}$ are a generalization of Smolyak sparse grids and allow for different dimensions to have different levels of refinement (Gerstner, Griebel, 2003).

To build the sparse grid, we follow the algorithm from Jakeman, Roberts (2011) and greedily add indexes and nodes with the largest approximation errors until a target accuracy is achieved. The algorithm adapts by both dimension and locality.

Define $\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}} = \left(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d}\right)$, $\psi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x}) = \psi_{j_1}^{i_1}(x_1) \cdots \psi_{j_d}^{i_d}(x_d)$, $\Delta X^0 = X^0$, and $\Delta X^i = X^i \setminus X^{i-1}$.

Algorithm 5 Build an interpolating polynomial on a sparse grid to approximate a function $f(\cdot)$ on $[0,1]^d$. Continue to refine the sparse grid until errors, specified by a function $f_\varepsilon(\cdot)$, are within a target tolerance.

function approximate(f, f_ε)
  $tol \leftarrow$ tolerance
  $G \leftarrow \{\}$ ▷ Subgrids and surpluses for the sparse grid
  $\mathcal{F} \leftarrow \{\}$ ▷ Expanded subgrids not yet added to G
  $\boldsymbol{i} \leftarrow \mathbf{0}$
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{\text{expand-subgrid}(G, f, f_\varepsilon, \boldsymbol{i})\}$
  while 1 **do**
    $\boldsymbol{i}, \varepsilon_{\boldsymbol{j}}^{\boldsymbol{i}} \leftarrow$ pick $\boldsymbol{i}$, $\varepsilon_{\boldsymbol{j}}^{\boldsymbol{i}}$ to maximize $\varepsilon_{\boldsymbol{j}}^{\boldsymbol{i}}$ in $\mathcal{F}$
    if $\varepsilon_{\boldsymbol{j}}^{\boldsymbol{i}} < tol$ **then**
      return G
    end if
    $G \leftarrow G \cup \left\{(\boldsymbol{i}, \boldsymbol{z}^{\boldsymbol{i}}, \boldsymbol{\varepsilon}^{\boldsymbol{i}})\right\}$
    $\mathcal{F} \leftarrow \mathcal{F} \setminus \left\{(\boldsymbol{i}, \boldsymbol{z}^{\boldsymbol{i}}, \boldsymbol{\varepsilon}^{\boldsymbol{i}})\right\}$
    for $\boldsymbol{i}_{\text{fwd}}$ in $\{\boldsymbol{i} + \boldsymbol{e}_k \mid 1 \le k \le d\}$ **do**
      if for all k such that $(\boldsymbol{i}_{\text{fwd}})_k > 0$, $\boldsymbol{i}_{\text{fwd}} - \boldsymbol{e}_k$ is in G **then**
        $\mathcal{F} \leftarrow \mathcal{F} \cup \{\text{expand-subgrid}(G, f, f_\varepsilon, \boldsymbol{i}_{\text{fwd}})\}$
      end if
    end for
  end while
end function
function expand-subgrid(G, f, f_ε, $\boldsymbol{i}$)
    ▷ Compute surpluses and errors for every active refinement node of the subgrid defined by $\boldsymbol{i}$
  $\boldsymbol{z}^{\boldsymbol{i}} \leftarrow \mathbf{0}$
  $\boldsymbol{\varepsilon}^{\boldsymbol{i}} \leftarrow \mathbf{0}$
  for $\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}}$ in $\Delta X^{i_1} \otimes \cdots \otimes \Delta X^{i_d}$ **do**
    if is-active(G, $\boldsymbol{i}$, $\boldsymbol{j}$) or $\boldsymbol{i} = \mathbf{0}$ **then**
      $y \leftarrow f(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}})$
      $\tilde{y} \leftarrow \text{evaluate}(G, \boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}})$
      $z_{\boldsymbol{j}}^{\boldsymbol{i}} \leftarrow y - \tilde{y}$
      $\varepsilon_{\boldsymbol{j}}^{\boldsymbol{i}} \leftarrow f_\varepsilon(y, \tilde{y}, \boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}})$
    end if
  end for
  return $\left(\boldsymbol{i}, \boldsymbol{z}^{\boldsymbol{i}}, \boldsymbol{\varepsilon}^{\boldsymbol{i}}\right)$
end function

Algorithm 5 (continued)

function evaluate(G, $\boldsymbol{x}$)
  ▷ Evaluate at $\boldsymbol{x}$ the polynomial interpolating the sparse grid G
  $res \leftarrow 0$
  for $\boldsymbol{i}, \boldsymbol{z}^{\boldsymbol{i}}$ in G **do**
    $res \leftarrow res + \sum_{z_{\boldsymbol{j}}^{\boldsymbol{i}} \in \boldsymbol{z}^{\boldsymbol{i}}} z_{\boldsymbol{j}}^{\boldsymbol{i}} \psi_{\boldsymbol{j}}^{\boldsymbol{i}}(\boldsymbol{x})$
  end for
  return res
end function
function is-active(G, $\boldsymbol{i}$, $\boldsymbol{j}$)
  ▷ Determine if the refinement $x_{\boldsymbol{j}}^{\boldsymbol{i}}$ is *active*. A refinement is active if it has at least one neighbor with an error that exceeds the cutoff threshold.
  $\tau \leftarrow$ cutoff threshold
  for k such that $i_k > 0$ **do**
    $\boldsymbol{i}_{\text{bwd}} \leftarrow \boldsymbol{i} - \boldsymbol{e}_k$
    for $\varepsilon_{\boldsymbol{j}_{\text{bwd}}}^{\boldsymbol{i}_{\text{bwd}}}$ in G **do**
      if $\varepsilon_{\boldsymbol{j}_{\text{bwd}}}^{\boldsymbol{i}_{\text{bwd}}} > \tau$ and is-point-neighbor($\boldsymbol{i}_{\text{bwd}}$, $\boldsymbol{j}_{\text{bwd}}$, $\boldsymbol{j}$, k) **then**
        return 1
      end if
    end for
  end for
  return 0
end function
function is-point-neighbor($\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{j}'$, k)
  ▷ Determine if the refinement $x_{\boldsymbol{j}'}^{\boldsymbol{i}+\boldsymbol{e}_k}$ neighbors the point $x_{\boldsymbol{j}}^{\boldsymbol{i}}$. See §4.2 of Jakeman, Roberts (2011) for details.
  if there exists $k' \neq k$ such that $j_{k'} \neq j'_{k'}$ **then**
    return 0
  else if $i_k \leq 1$ **then**
    return 1
  else
    return $\left(x_{j_k-1}^{i_k} < x_{j'_k}^{i_k+1} < x_{j_k}^{i_k}\right)$ or $\left(x_{j_k}^{i_k} < x_{j'_k}^{i_k+1} < x_{j_k+1}^{i_k}\right)$
  end if
end function

At $\mathbf{u}_{\text{map}}$, a second-order Taylor approximation to $f(\cdot)$ (7) gives us

$$f(\mathbf{u}_{\text{map}} + \boldsymbol{\delta}) \approx f(\mathbf{u}_{\text{map}}) + \frac{1}{2}\boldsymbol{\delta}'(\nabla^2 f)(\mathbf{u}_{\text{map}})\boldsymbol{\delta}.$$

If we use the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ as a basis, we have

$$f(\mathbf{u}_{\text{map}} + \delta_1\mathbf{v}_1 + \delta_2\mathbf{v}_2) \approx f(\mathbf{u}_{\text{map}}) + \frac{1}{2}\left(\xi_1\delta_1^2 + \xi_2\delta_2^2\right)$$

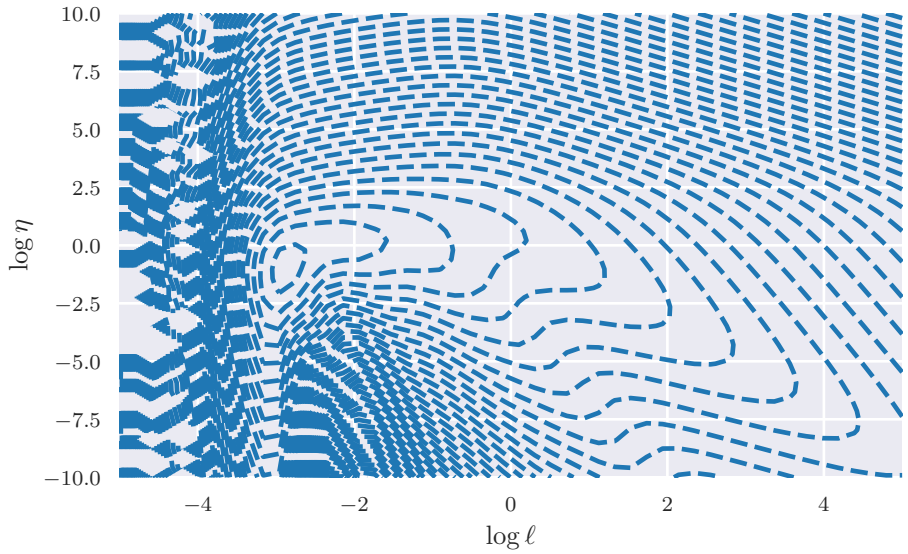

Figure 4: Reparameterized log posterior for the Example 1.1 data set with reference prior

where ξ_1 and ξ_2 are the eigenvalues of $\nabla^2 f(\mathbf{u}_{\text{map}})$. Thus, $\exp(f(\cdot))$ is approximately separable at $\mathbf{u}_{\text{map}}$ along the eigenvectors; and hence, $g(\cdot)$ (8) is approximately separable at $\mathbf{0.5}$,

$$g(0.5+\delta_1, 0.5+\delta_2) \approx h_1(0.5+\delta_1) \times h_2(0.5+\delta_2)$$

for some h_d and δ_d small. We can use this observation to build a more efficient approximation. Let $\tilde{h}_1(\cdot)$ and $\tilde{h}_2(\cdot)$ denote interpolations at Chebyshev nodes of the functions $g(\cdot, 0.5)$ and $g(0.5, \cdot)$. Then run Algorithm 5 with the target function $\frac{g(x_1,x_2)}{\tilde{h}_1(x_1)\tilde{h}_2(x_2)}$ and the error function $f_\varepsilon(y, \tilde{y}, x_1, x_2) = \left|(y-\tilde{y})\tilde{h}_1(x_1)\tilde{h}_2(x_2)\right|$.

> **Example 4.1.** (Example 1.2 continued) I ran Algorithm 5 on the data set from Example 1.1. Figure 4 shows contours for the log of the reparameterized posterior function, and Figure 5 shows the sparse grid used to approximate the reparameterized posterior.

Prediction Distributions

The sparse grid from Algorithm 5 naturally leads to a quadrature rule to approximate integration (Jakeman, Roberts, 2011). Let $f(\ell, \eta)$ denote a function.

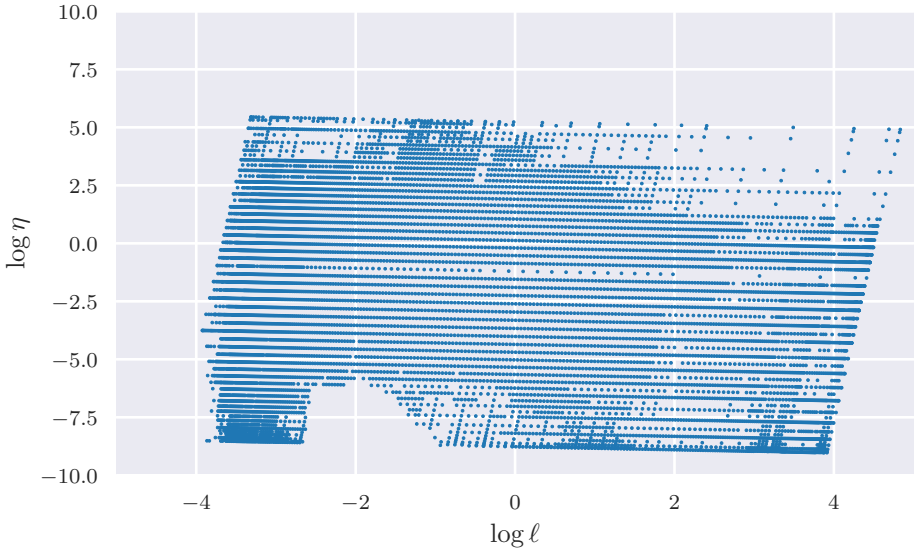

Figure 5: Sparse grid used to interpolate the reparameterized posterior for the Example 1.1 data set with reference prior

Put $\boldsymbol{\varphi}(\mathbf{u}) = (\varphi_\ell(u_1), \varphi_\eta(u_2))'$. Then

$$\begin{aligned}
&\int_0^\infty \int_0^\infty f(\ell,\eta)\,\pi\left(\ell,\eta \mid \mathbf{y}\right) d\ell d\eta \\
&\qquad \approx \int_0^\infty \int_0^\infty f(\ell,\eta)\,\tilde{\pi}\left(\ell,\eta \mid \mathbf{y}\right) d\ell d\eta \\
&\qquad \approx \frac{1}{Z}\int_0^1 \int_0^1 f\left(\boldsymbol{\varphi}(\mathbf{u}_{\text{map}} + s_1(x_1)\mathbf{v}_1 + s_2(x_2)\mathbf{v}_2)\right) \\
&\qquad\qquad\qquad\qquad g(x_1,x_2)\dot{s}_1(x_1)\dot{s}_2(x_2)dx_1dx_2 \\
&\qquad \approx \sum_k w_k f(\ell_k,\eta_k) \qquad (9)
\end{aligned}$$

where the points $\{(\ell_k,\eta_k)'\}$ are the transformed nodes of the sparse grid and weights are derived from integrals of the basis functions with the separable approximations,

$$\int_0^1 \psi_j^i(x)\tilde{h}_k(x)dx$$

for $k = 1, 2$.

Let $\tilde{\boldsymbol{s}}$ denote unobserved locations. Then

$$\mathrm{P}^\pi\left(Z(\tilde{s}_1),\ldots,Z(\tilde{s}_m) \mid \mathbf{y}\right) \approx \sum_k w_k\,\mathrm{P}^\pi\left(Z(\tilde{s}_1),\ldots,Z(\tilde{s}_m) \mid \mathbf{y},\ell_k,\eta_k\right)$$

gives us an approximation of the prediction distribution. Let's derive a more explicit formula for the conditional probability $\mathrm{P}^{\pi}(\cdot \mid \mathbf{y}, \ell, \eta)$. Use $\mathbf{y}_1 = \mathbf{y}$ to denote the observations and use $\mathbf{y}_2$ to denote possible values at the unobserved locations $\tilde{s}_1, \ldots, \tilde{s}_m$. Applying (3), we have

$$\begin{aligned}\mathrm{P}^{\pi}(\mathbf{y}_2 \mid \mathbf{y}_1, \ell, \eta) &\propto \int_0^{\infty} \int_{\mathbb{R}^p} \mathrm{P}(\mathbf{y}_1, \mathbf{y}_2 \mid \boldsymbol{\beta}, \sigma^2, \ell, \eta) \left(\frac{1}{\sigma^2}\right) d\boldsymbol{\beta} d\sigma^2 \\ &\propto \left[(\mathbf{y}_1, \mathbf{y}_2)\, \boldsymbol{R}\, (\mathbf{y}_1, \mathbf{y}_2)'\right]^{-(n+m-p)/2},\end{aligned}$$

where $\boldsymbol{R}$ is given by (4). Put $\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{12}' & \boldsymbol{R}_{22} \end{pmatrix}$. Then

$$\begin{aligned}(\mathbf{y}_1, \mathbf{y}_2)\, \boldsymbol{R}\, (\mathbf{y}_1, \mathbf{y}_2)' &= \mathbf{y}_1' \boldsymbol{R}_{11} \mathbf{y}_1 + 2\mathbf{y}_1' \boldsymbol{R}_{12} \mathbf{y}_2 + \mathbf{y}_2' \boldsymbol{R}_{22} \mathbf{y}_2 \\ &= (\mathbf{y}_2 - \bar{\mathbf{y}}_2)' \boldsymbol{R}_{22} (\mathbf{y}_2 - \bar{\mathbf{y}}_2) + b\end{aligned}$$

where $\bar{\mathbf{y}}_2 = -\boldsymbol{R}_{22}^{-1} \boldsymbol{R}_{12}' \mathbf{y}_1$ and $b = \mathbf{y}_1' \boldsymbol{R}_{11} \mathbf{y}_1 - \bar{\mathbf{y}}_2' \boldsymbol{R}_{22} \bar{\mathbf{y}}_2$.

σ^2 Marginal

The marginal distribution of σ^2 is given by

$$\begin{aligned}\mathrm{P}^{\pi}\left(\sigma^2 \mid \mathbf{y}\right) &= \int_0^{\infty} \int_0^{\infty} \mathrm{P}^{\pi}\left(\sigma^2 \mid \mathbf{y}, \ell, \eta\right) \pi(\ell, \eta \mid \mathbf{y})\, d\ell d\eta \\ &\approx \sum_k w_k\, \mathrm{P}^{\pi}\left(\sigma^2 \mid \mathbf{y}, \ell_k, \eta_k\right),\end{aligned}$$

where $\{w_k\}$, $\{\ell_k\}$, and $\{\eta_k\}$ are defined in (9). From (3), we have

$$\begin{aligned}\mathrm{P}^{\pi}\left(\sigma^2 \mid \mathbf{y}, \ell, \eta\right) &\propto \int_{\mathbb{R}^p} L(\boldsymbol{\beta}, \sigma^2, \ell, \eta; \mathbf{y})\, \pi\left(\boldsymbol{\beta}, \sigma^2 \mid \ell, \eta\right) d\boldsymbol{\beta} \\ &\propto \left(\sigma^2\right)^{-(n-p)/2} |\mathbf{G}|^{-1/2} |\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}|^{-1/2} \exp\left\{-\frac{S^2}{2\sigma^2}\right\} \left(\frac{1}{\sigma^2}\right) \\ &\propto \left(\frac{1}{\sigma^2}\right)^{(n-p)/2+1} \exp\left\{-\frac{S^2}{2\sigma^2}\right\}. \qquad (10)\end{aligned}$$

(10) is the unnormalized PDF of an inverse-gamma distribution. Normalizing gives us

$$\mathrm{P}^{\pi}\left(\sigma^2 \mid \mathbf{y}, \ell, \eta\right) = \frac{\left(S^2/2\right)^{(n-p)/2}}{\Gamma((n-p)/2)} \left(\frac{1}{\sigma^2}\right)^{(n-p)/2+1} \exp\left\{-\frac{S^2}{2\sigma^2}\right\}.$$

$\boldsymbol{\beta}$ Marginals

Similarly, to compute the posterior distribution of a particular regressor β_j, we have

$$\mathrm{P}^{\pi}(\beta_j \mid \mathbf{y}) = \int_0^{\infty} \int_0^{\infty} \mathrm{P}^{\pi}(\beta_j \mid \mathbf{y}, \ell, \eta)\, \pi(\ell, \eta \mid \mathbf{y})\, d\ell d\eta$$

where

$$\begin{aligned}
\mathrm{P}^{\pi}\left(\beta_j \mid \mathbf{y}, \ell, \eta\right) &\propto \int_0^{\infty} \int_{\mathbb{R}^{p-1}} \left(\frac{1}{\sigma^2}\right)^{n/2+1} \\
&\qquad \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\mathbf{G}^{-1}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right\} d\boldsymbol{\beta}_{/j} d\sigma^2 \\
&\propto \int_0^{\infty} \int_{\mathbb{R}^{p-1}} \left(\frac{1}{\sigma^2}\right)^{n/2+1} \\
&\qquad \exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}}\right)'\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}\left(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}}\right)\right\} \\
&\qquad \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{G}^{-1}\mathbf{y}-\bar{\boldsymbol{\beta}}\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}\bar{\boldsymbol{\beta}}\right)\right\} d\boldsymbol{\beta}_{/j} d\sigma^2 \\
&\propto \int_0^{\infty} \left(\frac{1}{\sigma^2}\right)^{(n-p+1)/2+1} \exp\left\{-\frac{1}{2\sigma^2}\frac{1}{\left(\boldsymbol{A}^{-1}\right)_{jj}}\left(\beta_j-\bar{\beta}_j\right)^2\right\} \\
&\qquad\qquad \exp\left\{-\frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{G}^{-1}\mathbf{y}-\bar{\boldsymbol{\beta}}\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}\bar{\boldsymbol{\beta}}\right)\right\} d\sigma^2 \\
&\propto \int_0^{\infty} \left(\frac{1}{\sigma^2}\right)^{(n-p+1)/2+1} \exp\left\{-\frac{1}{2\sigma^2}\#1\right\} d\sigma^2 \\
&\propto (\#1)^{-(n-p+1)/2}
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{A} &= \mathbf{X}'\mathbf{G}^{-1}\mathbf{X} \\
\bar{\beta} &= \boldsymbol{A}^{-1}\mathbf{X}'\mathbf{G}^{-1}\mathbf{y} \\
\#1 &= \frac{1}{\left(\boldsymbol{A}^{-1}\right)_{jj}}\left(\beta_j-\bar{\beta}_j\right)^2 + \mathbf{y}'\mathbf{G}^{-1}\mathbf{y} - \bar{\boldsymbol{\beta}}'\boldsymbol{A}\bar{\boldsymbol{\beta}} \\
&= \frac{1}{\left(\boldsymbol{A}^{-1}\right)_{jj}}\left(\beta_j-\bar{\beta}_j\right)^2 + S^2.
\end{aligned}$$

We recognize $\mathrm{P}^{\pi}\left(\beta_j \mid \mathbf{y}\right)$ as being a t-distribution with $n-p$ degrees of freedom, mean $\boldsymbol{\beta}_j$, and scale $s_{\beta_j} = \left\{\frac{\left(\boldsymbol{A}^{-1}\right)_{jj} S^2}{n-p}\right\}^{1/2}$.

ℓ, η Marginals

Algorithm 5 gives us an interpolating function that is inexpensive to evaluate and accurately approximates the reparameterized posterior $\pi\left(u_1, u_2 \mid \mathbf{y}\right) = \pi\left(\varphi_\ell(u_1), \varphi_\eta(u_2) \mid \mathbf{y}\right)\dot{\varphi}_\ell(u_1)\dot{\varphi}_\eta(u_2)$. Now,

$$\begin{aligned}
\pi\left(u_1 \mid \mathbf{y}\right) &= \int \pi\left(u_1, u_2 \mid \mathbf{y}\right) du_2 \\
&\approx \sum_k w_k\, \tilde{\pi}\left(u_1, t_k \mid \mathbf{y}\right),
\end{aligned}$$

where $\{w_k\}$ and $\{t_k\}$ can be chosen by the Gauss-Legendre quadrature rule for the interval determined by the bracketing region in Step 2 of Algorithm 3. If we evaluate $\tilde{\pi}(u_1 \mid \mathbf{y})$ at Chebyshev nodes across the range of u_1 in the bracketing region, then we obtain a polynomial that approximates $\pi(u_1 \mid \mathbf{y})$. $\pi(u_2 \mid \mathbf{y})$ can be similarly approximated using Chebyshev nodes.

If the error bounds from Step 2 are tight enough, the polynomial approximations for $\pi(u_1 \mid \mathbf{y})$ and $\pi(u_2 \mid \mathbf{y})$ will be suitable for estimating the CDFs. But since they cut off integration outside of the bracketing region, they won't accurately capture endpoint behavior and shouldn't be used for estimating moments. For example, $\pi(\eta \mid \mathbf{y})$ has infinite mean, which won't be reflected by the polynomial approximation.

5 Real Data Analysis

Let's apply the algorithms from §4 to real data.

5.1 Soil Carbon-to-Nitrogen

We'll first look at a data set from Schabenberger, Pierce (2001) of carbon-to-nitrogen ratios sampled across an agricultural field before and after tillage. The after-tillage data was analyzed by Ren et al. (2012) and De Oliveira (2007) using random sampling algorithms and a Gaussian process of the form (1) with

$$\mathbb{E}\{Z(\boldsymbol{s})\} = \beta_1 \quad \text{and} \quad \psi_\ell(d) = \exp\left\{-\frac{d}{\ell}\right\}.$$

We'll use the same model and data set with our deterministic algorithms. When we fit a sparse grid to approximate the posterior and marginalize, we get these values for the medians

$$(\beta_1)_{\text{med}} = 10.86, \quad \ell_{\text{med}} = 62.54, \quad \eta_{\text{med}} = 0.44, \quad \text{and} \quad \sigma^2_{\text{med}} = 0.24.$$

Figure 6 plots the sparse grid constructed by Algorithm 5; Figure 7 plots the posterior marginalizations for β_1, ℓ, η, and σ^2; and Figure 8 plots carbon-to-nitrogen predictions and credible sets across the agricultural field. [source]

5.2 Meuse River

Next, we'll look at a data set from the sp R-library containing 155 measurements of zinc concentration (ppm) collected in a flood plain of the river Meuse (Pebesma, Bivand, 2005). The data was previously analyzed by Kazianka, Pilz (2012) using a Gaussian process with a sampling algorithm. We'll use a similar model but with the deterministic algorithms from §4. We model log zinc concentration as a Gaussian process of the form (1) with

$$\mathbb{E}\{Z(\boldsymbol{s})\} = \beta_1 + \beta_2 x_1(\boldsymbol{s}) \quad \text{and} \quad \psi_\ell(d) = \exp\left\{-\frac{d}{\ell}\right\}$$

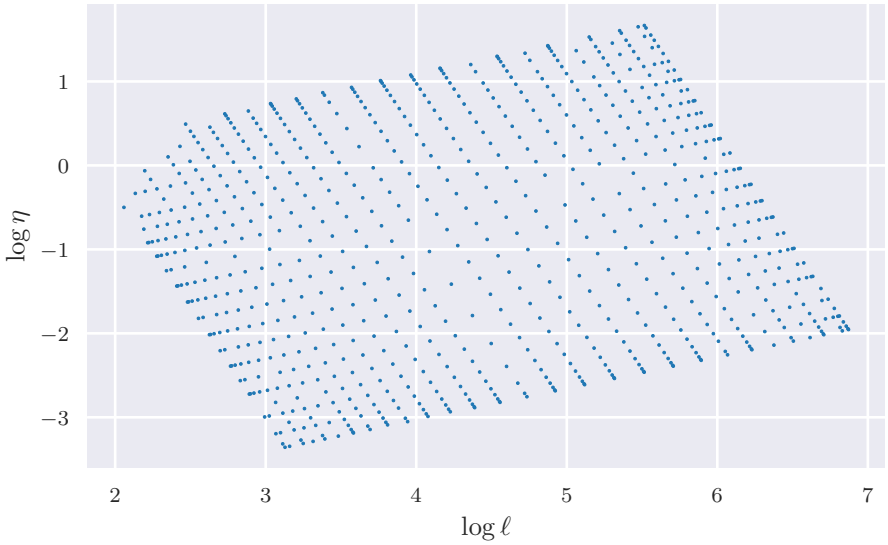

Figure 6: Sparse grid used to interpolate the posterior of the soil data set

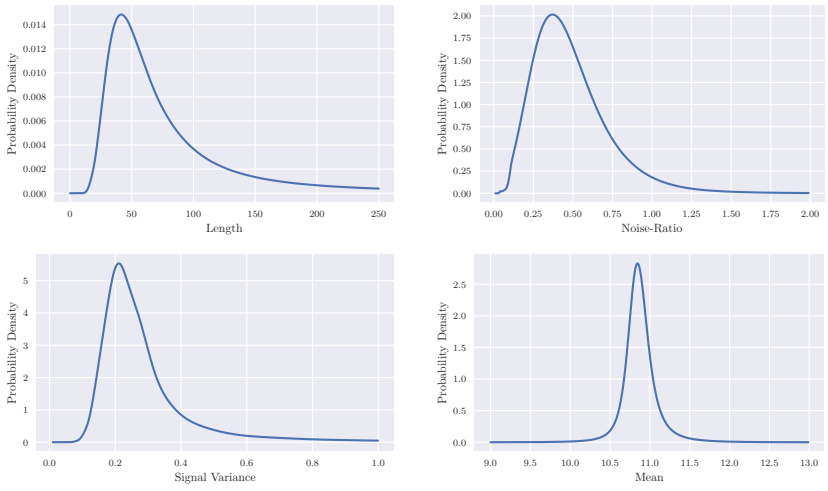

Figure 7: Marginalizations of the posterior distribution of the soil data set

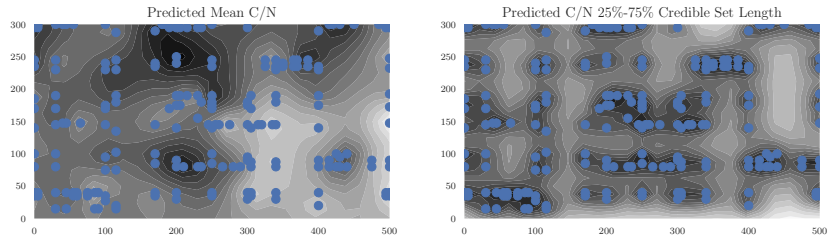

Figure 8: Prediction means and credible sets of soil carbon-to-nitrogen ratios for soil data set with sampling locations

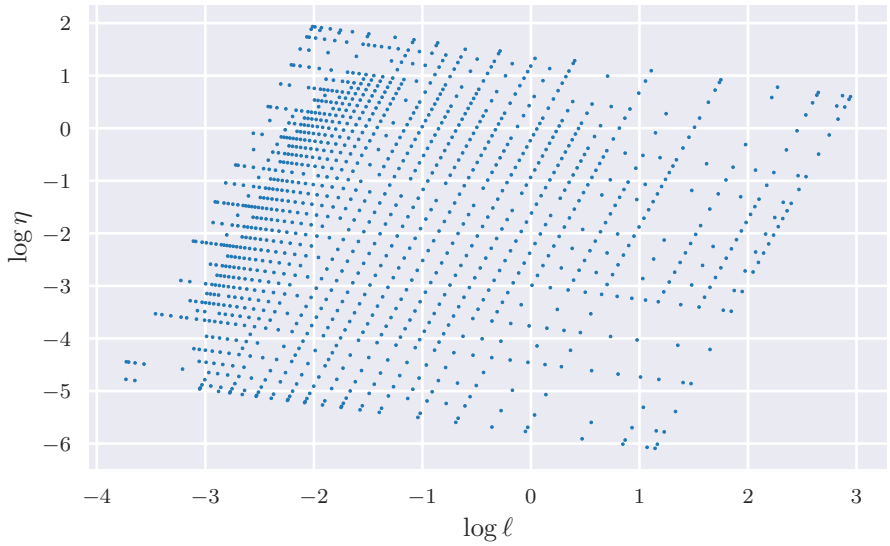

Figure 9: Sparse grid used to interpolate the posterior of the Meuse data set

where $x_1(\boldsymbol{s})$ is the square root of the distance of the flood plain sampling location, $\boldsymbol{s}$, to the river Meuse. After fitting the model, we compute medians

$$(\beta_1)_{\text{med}} = 6.99,\ (\beta_2)_{\text{med}} = -2.56,\ \ell_{\text{med}} = 0.22,\ \eta_{\text{med}} = 0.31,\ \text{and}\ \sigma^2_{\text{med}} = 0.16.$$

Figure 9 plots the sparse grid constructed by Algorithm 5; Figure 10 plots the posterior marginalizations for β_1, β_2, ℓ, η, and σ^2; and Figure 11 plots log zinc predictions and credible sets across the flood plain. [source]

5.3 Performance Analysis

To get a sense of the cost of the algorithms, I measured how long it took to apply the steps of Algorithm 3 to the soil and Meuse data sets for varying

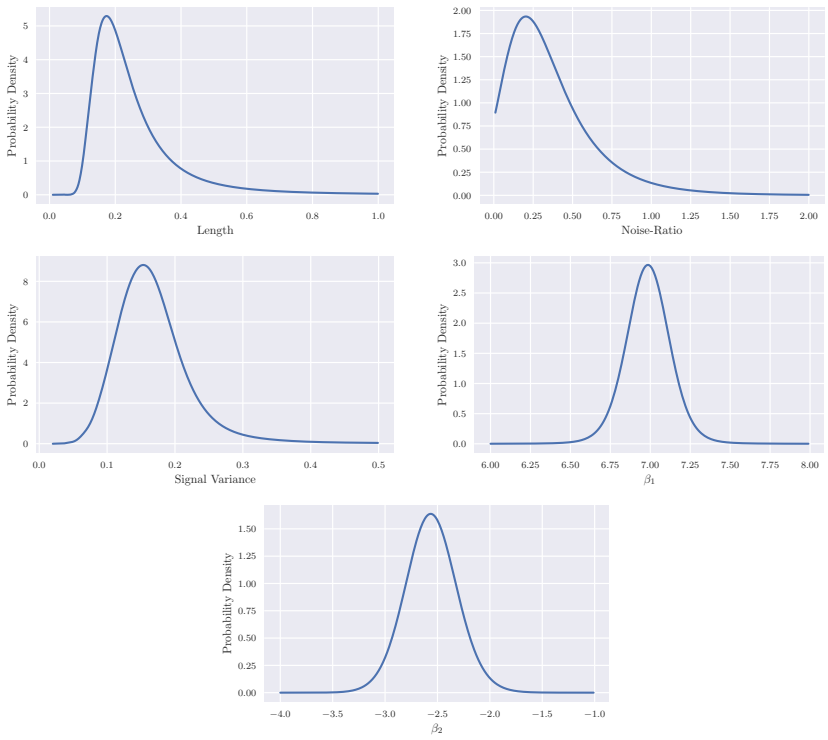

Figure 10: Marginalizations of the posterior distribution of the Meuse data set

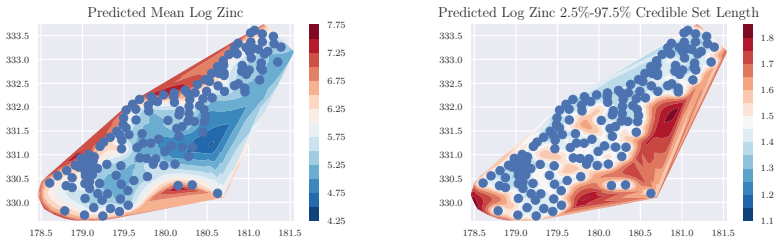

Figure 11: Prediction means and credible sets of log zinc concentration for Meuse data set with sampling locations

error tolerances. I computed the results using an 8-core AMD Ryzen 9 laptop. Table 10 and Table 11 summarize the performance results and provide the 25th, 50th, and 75th percentiles for the ℓ, η, and σ^2 distributions to help show how the tolerance affects accuracy. [source]

			ℓ Percentile			η Percentile			σ^2 Percentile		
tol	grid size	elapse (s)	25th	50th	75th	25th	50th	75th	25th	50th	75th
1×10^{-2}	249	1.37	42.00	63.50	106.92	0.31	0.45	0.60	0.20	0.25	0.31
1×10^{-3}	252	1.32	42.88	62.55	104.56	0.32	0.44	0.61	0.20	0.25	0.32
1×10^{-4}	798	2.86	42.88	62.55	104.56	0.32	0.44	0.61	0.20	0.25	0.32
1×10^{-5}	2218	6.67	42.88	62.54	104.57	0.32	0.44	0.61	0.20	0.25	0.32
1×10^{-6}	3415	9.82	42.88	62.54	104.57	0.32	0.44	0.61	0.20	0.25	0.32

Table 10: Provide distribution percentiles and measure elapse time for fitting a sparse grid to the soil data set described in §5.1 using various tolerances

			ℓ Percentile			η Percentile			σ^2 Percentile		
tol	grid size	elapse (s)	25th	50th	75th	25th	50th	75th	25th	50th	75th
1×10^{-2}	215	0.69	0.17	0.22	0.30	0.17	0.31	0.50	0.13	0.16	0.20
1×10^{-3}	416	1.03	0.17	0.22	0.30	0.17	0.31	0.49	0.13	0.16	0.20
1×10^{-4}	1193	2.42	0.17	0.22	0.30	0.17	0.31	0.50	0.13	0.16	0.20
1×10^{-5}	2358	4.51	0.17	0.22	0.30	0.17	0.31	0.50	0.13	0.16	0.20
1×10^{-6}	8666	17.10	0.17	0.22	0.30	0.17	0.31	0.50	0.13	0.16	0.20

Table 11: Provide distribution percentiles and measure elapse time for fitting a sparse grid to the Meuse data set described in §5.2 using various tolerances

6 Discussion

I presented deterministic algorithms for fully Bayesian prediction and inference for spatial models. In comparison to sampling methods such as MCMC, I would expect the algorithms to provide more reproducible results and require less tuning, making a more turnkey approach to analysis possible.

An area of future work could be to extend the algorithms to handle the Gaussian process models used in model emulation and calibration. In contrast to spatial Gaussian process models, models for emulation and calibration typically assume each dimension of the input space has a different scale and use a product covariance function (Sacks et al., 1989; Gu, 2019),

$$\operatorname{cov}\{Z(\boldsymbol{s}), Z(\mathbf{u})\} = \sigma^2 \prod_{k=1}^{d} \psi_{\ell_k}(\mid s_k - u_k \mid).$$

Paulo (2005) and Gu, Berger (2016) derived reference priors for separable covariance functions with distinct length parameters. Provided d is not too large, it may be possible to adopt the algorithms from §4 to work for this case to achieve deterministic fully Bayesian results. Additionally, certain modifications, such as using second-order information at interpolation points or assuming that the posterior is separable along the eigenvectors of its optimum's Hessian, could make the algorithms more efficient and larger values of d possible.

A Appendix: Posterior Derivatives

We will derive equations to compute the value, gradient, and Hessian of the negative log posterior $\pi(\ell, \eta \mid \mathbf{y})$. From (3) and (5), we have

$$\begin{aligned}\pi(\ell, \eta \mid \mathbf{y}) &\propto L^I(\ell, \eta; \mathbf{y}) \times \pi(\ell, \eta) \\ &\propto |\mathbf{G}|^{-1/2} |\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}|^{-1/2} \left(S^2\right)^{-(n-p)/2} |\boldsymbol{\Sigma}|^{1/2}.\end{aligned}$$

Put $\phi_1 = \ell$, $\phi_2 = \eta$, $\boldsymbol{A} = \mathbf{X}'\mathbf{G}^{-1}\mathbf{X}$, and define

$$f(\boldsymbol{\phi}) = \frac{1}{2}\log|\mathbf{G}| + \frac{1}{2}\log|\boldsymbol{A}| + \frac{n-p}{2}\log S^2 - \frac{1}{2}\log|\boldsymbol{\Sigma}|.$$

Let $\boldsymbol{L}_G$ denote the Cholesky factorization of $\mathbf{G}$, $\mathbf{G} = \boldsymbol{L}_G \boldsymbol{L}_G'$. Then

$$\begin{aligned}\boldsymbol{A} &= \mathbf{X}' \left(\boldsymbol{L}_G \boldsymbol{L}_G'\right)^{-1} \mathbf{X} \\ &= \mathbf{X}' \boldsymbol{L}_G'^{-1} \boldsymbol{L}_G^{-1} \mathbf{X} \\ &= \left(\boldsymbol{L}_G^{-1}\mathbf{X}\right)' \left(\boldsymbol{L}_G^{-1}\mathbf{X}\right).\end{aligned}$$

Let $\mathbf{Q}$ and $\boldsymbol{R}_A$ denote the QR factorization of $\boldsymbol{L}_G^{-1}\mathbf{X}$. Then

$$\begin{aligned}\boldsymbol{A} &= \left(\boldsymbol{L}_G^{-1}\mathbf{X}\right)'\left(\boldsymbol{L}_G^{-1}\mathbf{X}\right)\\ &= \left(\mathbf{Q}\boldsymbol{R}_A\right)'\left(\mathbf{Q}\boldsymbol{R}_A\right)\\ &= \boldsymbol{R}_A'\mathbf{Q}'\mathbf{Q}\boldsymbol{R}_A\\ &= \boldsymbol{R}_A'\boldsymbol{R}_A.\end{aligned}$$

Put $\mathbf{H} = \mathbf{G}^{-1}\mathbf{X}\boldsymbol{A}^{-1}\mathbf{X}'\mathbf{G}^{-1}$ and $\boldsymbol{F}_H = \boldsymbol{R}_A'^{-1}\mathbf{X}'\mathbf{G}^{-1}$. Applying to $\boldsymbol{R}$ (4), we have $\mathbf{H} = \boldsymbol{F}_H'\boldsymbol{F}_H$ and $\boldsymbol{R} = \boldsymbol{L}_G'^{-1}\boldsymbol{L}_G^{-1} + \boldsymbol{F}_H'\boldsymbol{F}_H$.

Gradient

Put

$$\#1 = \frac{1}{2}\log|\mathbf{G}|, \quad \#2 = \frac{1}{2}\log|\boldsymbol{A}|, \quad \#3 = \frac{n-p}{2}\log S^2, \text{ and } \quad \#4 = \frac{1}{2}\log|\boldsymbol{\Sigma}|.$$

Applying Jacobi's formula, $\frac{d}{dt}|\boldsymbol{B}(t)| = |\boldsymbol{B}|\,\mathrm{tr}\left\{\boldsymbol{B}^{-1}\frac{d\boldsymbol{B}}{dt}\right\}$, we have

$$\begin{aligned}\frac{\partial \#1}{\partial \phi_s} &= \frac{1}{2}\,\mathrm{tr}\left\{\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \phi_s}\right\},\\ \frac{\partial \#2}{\partial \phi_s} &= \frac{1}{2}\,\mathrm{tr}\left\{\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial \phi_s}\right\},\\ \frac{\partial \#3}{\partial \phi_s} &= \frac{n-p}{2}\frac{1}{S^2}\frac{\partial S^2}{\partial \phi_s} = \frac{n-p}{2}\frac{1}{S^2}\mathbf{y}'\frac{\partial \boldsymbol{R}}{\partial \phi_s}\mathbf{y}, \quad \text{and}\\ \frac{\partial \#4}{\partial \phi_s} &= \frac{1}{2}\,\mathrm{tr}\left\{\boldsymbol{\Sigma}^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right\}.\end{aligned}$$

Using the formula for differentiating an inverse matrix, $\frac{d}{dt}\boldsymbol{B}(t)^{-1} = -\boldsymbol{B}^{-1}\frac{d\boldsymbol{B}}{dt}\boldsymbol{B}^{-1}$, we derive the derivative of $\boldsymbol{A}$,

$$\frac{\partial \boldsymbol{A}}{\partial \phi_s} = \mathbf{X}'\frac{\partial \mathbf{G}^{-1}}{\partial \phi_s}\mathbf{X}$$

where $\frac{\partial \mathbf{G}^{-1}}{\partial \phi_s} = -\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{G}^{-1}$. Differentiating $\boldsymbol{R}$ gives us

$$\frac{\partial \boldsymbol{R}}{\partial \phi_s} = \frac{\partial}{\partial \phi_s}\left(\mathbf{G}^{-1} - \mathbf{H}\right) = \frac{\partial \mathbf{G}^{-1}}{\partial \phi_s} - \frac{\partial \mathbf{H}}{\partial \phi_s}$$

and

$$\begin{aligned}
\frac{\partial \mathbf{H}}{\partial \phi_s} &= \frac{\partial}{\partial \phi_s}\left(\mathbf{G}^{-1}\mathbf{X}\boldsymbol{A}^{-1}\mathbf{X}'\mathbf{G}^{-1}\right) \\
&= -\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{H} - \mathbf{H}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{G}^{-1} - \mathbf{G}^{-1}\mathbf{X}\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial \phi_s}\boldsymbol{A}^{-1}\mathbf{X}\mathbf{G}^{-1} \\
&= -\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{H} - \mathbf{H}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{G}^{-1} \\
&\qquad + \mathbf{G}^{-1}\mathbf{X}\boldsymbol{A}^{-1}\left(\mathbf{X}'\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{G}^{-1}\mathbf{X}\right)\boldsymbol{A}^{-1}\mathbf{X}\mathbf{G}^{-1} \\
&= -\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{H} - \mathbf{H}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{G}^{-1} + \mathbf{H}\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{H} \\
&= -\left(\mathbf{G}^{-1} - \frac{1}{2}\mathbf{H}\right)\frac{\partial \mathbf{G}}{\partial \phi_s}\mathbf{H} - \mathbf{H}\frac{\partial \mathbf{G}}{\partial \phi_s}\left(\mathbf{G}^{-1} - \frac{1}{2}\mathbf{H}\right).
\end{aligned}$$

Put $\#5 = \operatorname{tr}\left\{\boldsymbol{R}\frac{\partial \mathbf{K}}{\partial \ell}\right\}$. Then

$$\begin{aligned}
\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right)_{11} &= \frac{\partial}{\partial \phi_s}\operatorname{tr}\left\{\#5^2\right\} = 2\operatorname{tr}\left\{\#5\frac{\partial \#5}{\partial \phi_s}\right\}, \\
\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right)_{12} &= \frac{\partial}{\partial \phi_s}\operatorname{tr}\left\{\boldsymbol{R}^2\frac{\partial \mathbf{K}}{\partial \ell}\right\} = \operatorname{tr}\left\{\frac{\partial \boldsymbol{R}^2}{\partial \phi_s}\frac{\partial \mathbf{K}}{\partial \ell} + \boldsymbol{R}^2\frac{\partial^2 \mathbf{K}}{\partial \phi_s \partial \ell}\right\}, \\
\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right)_{13} &= \operatorname{tr}\left\{\frac{\partial \#5}{\partial \phi_s}\right\}, \\
\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right)_{22} &= \operatorname{tr}\left\{\frac{\partial \boldsymbol{R}^2}{\partial \phi_s}\right\}, \\
\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right)_{23} &= \operatorname{tr}\left\{\frac{\partial \boldsymbol{R}}{\partial \phi_s}\right\}, \\
\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_s}\right)_{33} &= 0,
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \#5}{\partial \phi_s} &= \frac{\partial}{\partial \phi_s}\left(\boldsymbol{R}\frac{\partial \mathbf{K}}{\partial \ell}\right) = \frac{\partial \boldsymbol{R}}{\partial \phi_s}\frac{\partial \mathbf{K}}{\partial \ell} + \boldsymbol{R}\frac{\partial^2 \mathbf{K}}{\partial \phi_s \partial \ell} \quad \text{and} \\
\frac{\partial \boldsymbol{R}^2}{\partial \phi_s} &= \frac{\partial \boldsymbol{R}}{\partial \phi_s}\boldsymbol{R} + \boldsymbol{R}\frac{\partial \boldsymbol{R}}{\partial \phi_s}.
\end{aligned}$$

Hessian

Computing second derivatives we have

$$\begin{aligned}
\frac{\partial^2 \#1}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{1}{2}\operatorname{tr}\left\{\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\right\}\right) \\
&= \frac{1}{2}\operatorname{tr}\left\{-\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_s}\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t} + \mathbf{G}^{-1}\frac{\partial^2\mathbf{G}}{\partial\phi_s\partial\phi_t}\right\}, \\
\frac{\partial^2 \#2}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{1}{2}\operatorname{tr}\left\{\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial\phi_t}\right\}\right) \\
&= \frac{1}{2}\operatorname{tr}\left\{-\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial\phi_s}\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial\phi_t} + \boldsymbol{A}^{-1}\frac{\partial^2\boldsymbol{A}}{\partial\phi_s\partial\phi_t}\right\}, \\
\frac{\partial^2 \#3}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{n-p}{2}\frac{1}{S^2}\frac{\partial S^2}{\partial\phi_t}\right) \\
&= \frac{n-p}{2}\left(-\frac{1}{S^4}\frac{\partial S^2}{\partial\phi_s}\frac{\partial S^2}{\partial\phi_t} + \frac{1}{S^2}\frac{\partial^2 S^2}{\partial\phi_s\partial\phi_t}\right), \quad \text{and} \\
\frac{\partial^2 \#4}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{1}{2}\operatorname{tr}\left\{\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{\Sigma}}{\partial\phi_t}\right\}\right) \\
&= \frac{1}{2}\operatorname{tr}\left\{-\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{\Sigma}}{\partial\phi_s}\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{\Sigma}}{\partial\phi_t} + \boldsymbol{\Sigma}^{-1}\frac{\partial^2\boldsymbol{\Sigma}}{\partial\phi_s\partial\phi_t}\right\}.
\end{aligned}$$

For the second derivative of $\boldsymbol{A}$, we have

$$\begin{aligned}
\frac{\partial^2\boldsymbol{A}}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(-\mathbf{X}'\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{G}^{-1}\mathbf{X}\right) \\
&= \mathbf{X}'\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_s}\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{G}^{-1}\mathbf{X} + \mathbf{X}'\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_s}\mathbf{G}^{-1}\mathbf{X} \\
&\qquad - \mathbf{X}'\mathbf{G}^{-1}\frac{\partial^2\mathbf{G}}{\partial\phi_s\partial\phi_t}\mathbf{G}^{-1}\mathbf{X}.
\end{aligned}$$

Differentiating $\boldsymbol{R}$ a second time gives us

$$\begin{aligned}
\frac{\partial^2\boldsymbol{R}}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{\partial\mathbf{G}^{-1}}{\partial\phi_t} - \frac{\partial\mathbf{H}}{\partial\phi_t}\right) \\
&= \frac{\partial^2\mathbf{G}^{-1}}{\partial\phi_s\partial\phi_t} - \frac{\partial^2\mathbf{H}}{\partial\phi_s\partial\phi_t}
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial^2 \mathbf{G}^{-1}}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(-\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{G}^{-1}\right) \\
&= \mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_s}\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{G}^{-1} + \mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{G}^{-1}\frac{\partial\mathbf{G}}{\partial\phi_s}\mathbf{G}^{-1} \\
&\qquad - \mathbf{G}^{-1}\frac{\partial^2\mathbf{G}}{\partial\phi_s\partial\phi_t}\mathbf{G}^{-1} \quad \text{and} \\
\frac{\partial^2 \mathbf{H}}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(-\left(\mathbf{G}^{-1}-\frac{1}{2}\mathbf{H}\right)\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{H} - \mathbf{H}\frac{\partial\mathbf{G}}{\partial\phi_t}\left(\mathbf{G}^{-1}-\frac{1}{2}\mathbf{H}\right)\right) \\
&= \#D2H1 + \#D2H2 + \#D2H3
\end{aligned}$$

and

$$\begin{aligned}
\#D2H1 &= -\left(\frac{\partial\mathbf{G}^{-1}}{\partial\phi_s} - \frac{1}{2}\frac{\partial\mathbf{H}}{\partial\phi_s}\right)\frac{\partial\mathbf{G}}{\partial\phi_t}\mathbf{H} - \mathbf{H}\frac{\partial\mathbf{G}}{\partial\phi_t}\left(\frac{\partial\mathbf{G}^{-1}}{\partial\phi_s} - \frac{1}{2}\frac{\partial\mathbf{H}}{\partial\phi_s}\right), \\
\#D2H2 &= -\left(\mathbf{G}^{-1}-\frac{1}{2}\mathbf{H}\right)\frac{\partial\mathbf{G}}{\partial\phi_t}\frac{\partial\mathbf{H}}{\partial\phi_s} - \frac{\partial\mathbf{H}}{\partial\phi_s}\frac{\partial\mathbf{G}}{\partial\phi_t}\left(\mathbf{G}^{-1}-\frac{1}{2}\mathbf{H}\right), \quad \text{and} \\
\#D2H3 &= -\left(\mathbf{G}^{-1}-\frac{1}{2}\mathbf{H}\right)\frac{\partial^2\mathbf{G}}{\partial\phi_s\partial\phi_t}\mathbf{H} - \mathbf{H}\frac{\partial^2\mathbf{G}}{\partial\phi_s\partial\phi_t}\left(\mathbf{G}^{-1}-\frac{1}{2}\mathbf{H}\right).
\end{aligned}$$

Computing the second derivative of $\mathbf{\Sigma}$, we have

$$\begin{aligned}
\left(\frac{\partial^2\mathbf{\Sigma}}{\partial\phi_s\partial\phi_t}\right)_{11} &= \frac{\partial}{\partial\phi_s}\left(2\,\mathrm{tr}\left\{\#5\frac{\partial\#5}{\partial\phi_t}\right\}\right) \\
&= 2\,\mathrm{tr}\left\{\frac{\partial\#5}{\partial\phi_s}\frac{\partial\#5}{\partial\phi_t} + \#5\frac{\partial^2\#5}{\partial\phi_s\partial\phi_t}\right\}, \\
\left(\frac{\partial^2\mathbf{\Sigma}}{\partial\phi_s\partial\phi_t}\right)_{12} &= \frac{\partial}{\partial\phi_s}\left(\mathrm{tr}\left\{\frac{\partial\boldsymbol{R}^2}{\partial\phi_s}\frac{\partial\mathbf{K}}{\partial\ell} + \boldsymbol{R}^2\frac{\partial^2\mathbf{K}}{\partial\phi_t\partial\ell}\right\}\right) \\
&= \mathrm{tr}\left\{\frac{\partial^2\boldsymbol{R}^2}{\partial\phi_s\partial\phi_t}\frac{\partial\mathbf{K}}{\partial\ell} + \frac{\partial\boldsymbol{R}^2}{\partial\phi_s}\frac{\partial^2\mathbf{K}}{\partial\phi_t\partial\ell} + \frac{\partial\boldsymbol{R}^2}{\partial\phi_s}\frac{\partial^2\mathbf{K}}{\partial\phi_s\partial\ell} + \boldsymbol{R}^2\frac{\partial^3\mathbf{K}}{\partial\phi_s\partial\phi_t\partial\ell}\right\}, \\
\left(\frac{\partial^2\mathbf{\Sigma}}{\partial\phi_s\partial\phi_t}\right)_{13} &= \mathrm{tr}\left\{\frac{\partial\#5}{\partial\phi_s}\phi_t\right\}, \\
\left(\frac{\partial^2\mathbf{\Sigma}}{\partial\phi_s\partial\phi_t}\right)_{22} &= \mathrm{tr}\left\{\frac{\partial^2\boldsymbol{R}^2}{\partial\phi_s\partial\phi_t}\right\}, \quad \text{and} \\
\left(\frac{\partial^2\mathbf{\Sigma}}{\partial\phi_s\partial\phi_t}\right)_{23} &= \mathrm{tr}\left\{\frac{\partial^2\boldsymbol{R}}{\partial\phi_s\partial\phi_t}\right\}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial^2 \#5}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{\partial \boldsymbol{R}}{\partial\phi_t}\frac{\partial \mathbf{K}}{\partial\ell} + \boldsymbol{R}\frac{\partial^2 \mathbf{K}}{\partial\phi_t\partial\ell}\right) \\
&= \frac{\partial^2 \boldsymbol{R}}{\partial\phi_s\partial\phi_t}\frac{\partial \mathbf{K}}{\partial\ell} + \frac{\partial \boldsymbol{R}}{\partial\phi_s}\frac{\partial^2 \mathbf{K}}{\partial\phi_t\partial\ell} + \frac{\partial \boldsymbol{R}}{\partial\phi_t}\frac{\partial^2 \mathbf{K}}{\partial\phi_s\partial\ell} + \boldsymbol{R}\frac{\partial^3 \mathbf{K}}{\partial\phi_s\partial\phi_t\partial\ell} \quad \text{and} \\
\frac{\partial^2 \boldsymbol{R}^2}{\partial\phi_s\partial\phi_t} &= \frac{\partial}{\partial\phi_s}\left(\frac{\partial \boldsymbol{R}}{\partial\phi_t}\boldsymbol{R} + \boldsymbol{R}\frac{\partial \boldsymbol{R}}{\partial\phi_t}\right) \\
&= \left(\frac{\partial^2 \boldsymbol{R}}{\partial\phi_s\partial\phi_t}\boldsymbol{R} + \boldsymbol{R}\frac{\partial^2 \boldsymbol{R}}{\partial\phi_s\partial\phi_t}\right) + \left(\frac{\partial \boldsymbol{R}}{\partial\phi_s}\frac{\partial \boldsymbol{R}}{\partial\phi_t} + \frac{\partial \boldsymbol{R}}{\partial\phi_t}\frac{\partial \boldsymbol{R}}{\partial\phi_s}\right).
\end{aligned}$$

References

Barthelmann Volker, Novak Erich, Ritter Klaus. High dimensional polynomial interpolation on sparse grids // Advances in Computational Mathematics. 2000. 12. 273–288.

Berger J., Bernardo Jose. On the development of reference priors // Bayesian Stat. 11 1991. 4.

Berger James. The case for objective Bayesian analysis // Bayesian Analysis. 2006. 1, 3. 385 – 402.

Berger James O., Liseo Brunero, Wolpert Robert L. Integrated likelihood methods for eliminating nuisance parameters // Statistical Science. 1999. 14, 1. 1 – 28.

Berger James O, Oliveira Victor De, Sansó Bruno. Objective Bayesian Analysis of Spatially Correlated Data // Journal of the American Statistical Association. 2001. 96, 456. 1361–1374.

De Oliveira Victor. Objective Bayesian analysis of spatial data with measurement error // Canadian Journal of Statistics. 06 2007. 35. 283 – 301.

Fritsch F. N., Carlson R. E. Monotone Piecewise Cubic Interpolation // SIAM Journal on Numerical Analysis. 1980. 17, 2. 238–246.

Gerstner Thomas, Griebel Michael. Dimension–Adaptive Tensor–Product Quadrature // Computing. 09 2003. 71. 65–87.

Gu Mengyang. Jointly Robust Prior for Gaussian Stochastic Process in Emulation, Calibration and Variable Selection // Bayesian Analysis. 2019. 14, 3. 857 – 885.

Gu Mengyang, Berger James. Parallel partial Gaussian process emulation for computer models with massive output // The Annals of Applied Statistics. 09 2016. 10. 1317–1347.

Gu Mengyang, Wang Xiaojing, Berger James O. Robust Gaussian stochastic process emulation // The Annals of Statistics. 2018. 46, 6A. 3038 – 3066.

Jakeman John D., Roberts Stephen G. Local and Dimension Adaptive Sparse Grid Interpolation and Quadrature. 2011.

Kazianka Hannes, Pilz Jürgen. Objective Bayesian analysis of spatial data with uncertain nugget and range parameters // Canadian Journal of Statistics. 2012. 40.

Klimke Andreas. Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids. 01 2006. 40–41.

Moré Jorge J., Sorensen D. C. Computing a Trust Region Step // SIAM Journal on Scientific and Statistical Computing. 1983. 4, 3. 553–572.

Nocedal Jorge, Wright Stephen J. Numerical Optimization. New York, NY, USA: Springer, 2006. 2e. 92–93.

Paulo Rui. Default priors for Gaussian processes // The Annals of Statistics. 2005. 33, 2. 556 – 582.

Pebesma Edzer J., Bivand Roger S. Classes and methods for spatial data in R // R News. November 2005. 5, 2. 9–13.

Ren Cuirong, Sun Dongchu, He Chong. Objective Bayesian analysis for a spatial model with nugget effects // Journal of Statistical Planning and Inference. 2012. 142, 7. 1933–1946.

Sacks Jerome, Welch William J., Mitchell Toby J., Wynn Henry P. Design and Analysis of Computer Experiments // Statistical Science. 1989. 4, 4. 409 – 423.

Schabenberger Oliver, Pierce Fran. Contemporary Statistical Models for the Plant and Soil Science. 11 2001. 738.

Sorensen D. C. Newton's Method with a Model Trust Region Modification // SIAM Journal on Numerical Analysis. 1982. 19, 2. 409–426.

Trefethen Lloyd N. Approximation Theory and Approximation Practice, Extended Edition. USA: Society for Industrial and Applied Mathematics, 2019.

Welch B. L., Peers H. W. On Formulae for Confidence Points Based on Integrals of Weighted Likelihoods // Journal of the royal statistical society series b-methodological. 1963. 25. 318–329.