Deterministic Objective Bayesian Inference for Spatial Models

Ryan Burn
email ryan.burn@gmail.com

May 7, 2023

Abstract

Berger et al. (2001) and Ren et al. (2012) developed noninformative priors for Gaussian process regression based off of the reference prior approach (Berger, Bernardo, 1991). The priors have good statistical properties and provide a basis for objective Bayesian analysis (Berger, 2006). This paper and the project `https://github.com/rnburn/bbai` provide deterministic algorithms to do statistical inference using the priors.

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

$$\boldsymbol{\theta} = \left(\boldsymbol{\beta}, \sigma^2, \ell, \eta\right)'.$$

In order 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

$$\mathbf{y} = \left(Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_n)\right)'.$$

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, $\boldsymbol{\theta}_{\mathrm{ml}}$. Then $\boldsymbol{\theta}_{\mathrm{true}}$ should be close to $\boldsymbol{\theta}_{\mathrm{ml}}$, and

$$\mathrm{P}\left(Z(\mathbf{u}) \mid \mathbf{y}, \boldsymbol{\theta}_{\mathrm{ml}}\right)$$

should be a reasonable substitute for

$$\mathrm{P}\left(Z(\mathbf{u}) \mid \mathbf{y}, \boldsymbol{\theta}_{\mathrm{true}}\right).$$

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

Example 1.1. *Consider this data set.*

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*

I randomly sampled the Gaussian process

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

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

$$\sigma^2_{ml} = 34.42, \quad \ell_{ml} = 0.035, \quad \eta_{ml} = 8.27 \times 10^{-7}.$$

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

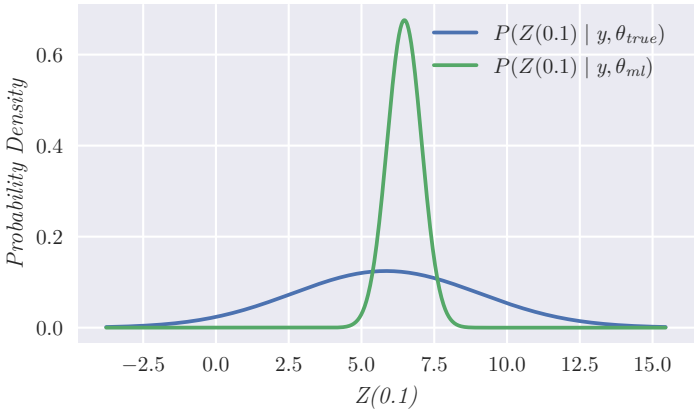

Figure 1: Compare prediction probability distributions when $\boldsymbol{\theta}_{\text{ml}}$ is used as a substitute for $\boldsymbol{\theta}_{\text{true}}$

Put

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

Figure 2 plots $g(t)$ for $0 \leq t \leq 1$. We see that likelihood is definitely not strongly peaked about the optimum, and any of the parameters along the line segment could have reasonably produced $\mathbf{y}$*. See here for source code.*

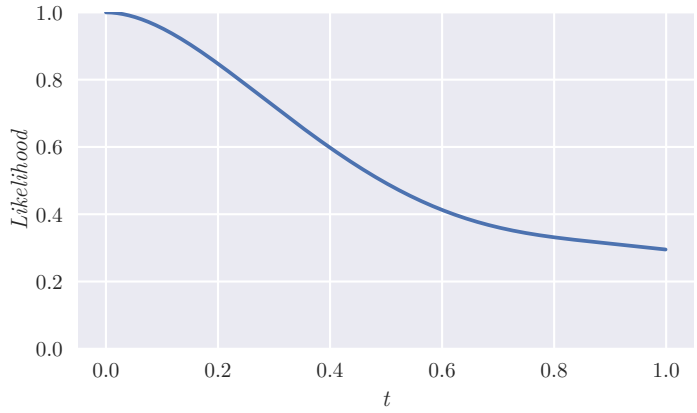

Figure 2: Likelihood for different values of $\boldsymbol{\theta}$ on the line segment from $\boldsymbol{\theta}_{\text{ml}}$ to $\boldsymbol{\theta}_{\text{true}}$

Approach 2: Integrate over Possible Parameters

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

single value 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 oberservations $\mathbf{y}$. 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 parameters $\boldsymbol{\theta}$ generate the data before seeing observations.

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 remain hotly 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 inference algorithm), let's look at how the approach works on the Gaussian process from Example 1.1.

Example 1.2. *(Example 1.1 continued) In Figure 3, I plot the prediction from Example 1.1 using the Bayesian approach with a reference prior. We see it's much closer to the true distribution than the distribution using maximum likelihood parameters. See here for source code.*

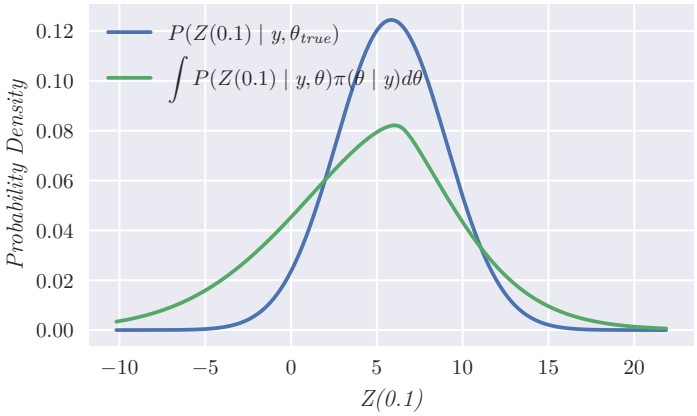

Figure 3: Compare prediction distribution from the Bayesian approach with reference prior to the prediction distribution of $\boldsymbol{\theta}_{\text{true}}$.

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 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 results close to α.

Algorithm 1 Test accuracy of credible sets produced with a prior

function COVERAGE-TEST($N, \tilde{\boldsymbol{\theta}}, j, \alpha$)
  $cnt \leftarrow 0$
  for $i \leftarrow 1$ to N **do**
    $\tilde{\mathbf{y}} \leftarrow$ sample from $P(\mathbf{y} \mid \tilde{\boldsymbol{\theta}})$
    $\tilde{\boldsymbol{\Theta}} \leftarrow \Theta_1 \times \cdots \times \Theta_{j-1} \times \Theta_j \cap (-\infty, \tilde{\theta}_j] \times \Theta_{j+1} \times \cdots \times \Theta_k$
    $t \leftarrow \int_{\tilde{\boldsymbol{\Theta}}} \pi(\boldsymbol{\theta} \mid \tilde{\mathbf{y}}) d\boldsymbol{\theta}$
    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

$$\pi(\boldsymbol{\theta}) \propto 1.$$

Immediately, we see one serious disadvantage of this approach: It's not invariant under reparameterization. If $\varphi(\cdot)$ is some monotonically increasing surjective function, then

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

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. *Suppose we observe n normally distributed values, $\mathbf{y}$, with variance 1 and unknown mean, μ. Then*

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

Thus,

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

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

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
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*

See here for source code.

Example 2.2. *Suppose we observe n normally distributed values, $\mathbf{y}$, with zero-mean and unknown variance, σ^2. Then*

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

where

$$s^2 = \frac{\mathbf{y}'\mathbf{y}}{n}.$$

Put

$$\sigma^2 = \frac{ns^2}{2u}.$$

Then

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

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. The results are below

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
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*

See here for source code.

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

$$\pi\left(\boldsymbol{\theta}\right) \propto |\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})|^{1/2},$$

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 U, then the change of variables formula gives us

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

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,

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

and

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

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

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

Then

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

Now, $y_1^2 + \cdots + y_n^2$ follows a chi-squared distribution and with variance $2n\sigma^4$ and 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

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

And 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 these coverage results:

$n=5$		$n=10$		$n=15$		$n=20$	
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*

See here for source code.

So far, Jeffreys prior performs excellently. In fact, for a single parameter, Welch, Peers (1963) showed 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. *Suppose we observe n normally distributed values, $\mathbf{y}$, with unknown mean, μ, and variance, σ^2. Then*

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

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

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

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

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

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.

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
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*

See here for source code.

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.

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, 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. *(Example 2.5 continued). We first integrate out μ using the constant conditional prior*

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

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

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

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

$$\begin{aligned} \mathbb{E}[Z] &= n-1 \\ \mathbb{E}[Z^2] &= 2(n-1) + (n-1)^2, \end{aligned}$$

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 these results:

$n = 5$		$n = 10$		$n = 15$		$n = 20$	
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*

See here for source code.

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

$$\begin{aligned} \mathbf{X} &= \left(\boldsymbol{x}(\boldsymbol{s}_1), \ldots, \boldsymbol{x}(\boldsymbol{s}_n)\right)', \\ \mathbf{G} &= \eta \boldsymbol{I} + \mathbf{K}(\ell), \\ \left(\mathbf{K}(\ell)\right)_{ij} &= \psi_\ell\left(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|\right). \end{aligned}$$

Integrating likelihood with the conditional prior gives us

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

where

$$S^2 = \mathbf{y}'\boldsymbol{R}\mathbf{y}$$
$$\boldsymbol{R} = \mathbf{G}^{-1} - \mathbf{G}^{-1}\mathbf{X}\left(\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{G}^{-1}. \tag{3}$$

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

$$\pi\left(\boldsymbol{\beta}, \sigma^2, \ell, \eta\right) \propto \left(\frac{1}{\sigma^2}\right)|\boldsymbol{\Sigma}(\ell,\eta)|^{1/2}, \tag{4}$$

where

$$\boldsymbol{\Sigma}(\ell,\eta) = \begin{pmatrix} \operatorname{tr}\left\{(\boldsymbol{R}\frac{\partial \mathbf{K}}{\partial \ell})^2\right\} & \operatorname{tr}(\boldsymbol{R}^2\frac{\partial \mathbf{K}}{\partial \ell}) & \operatorname{tr}(\boldsymbol{R}\frac{\partial \mathbf{K}}{\partial \ell}) \\ * & \operatorname{tr}(\boldsymbol{R}^2) & \operatorname{tr}(\boldsymbol{R}) \\ * & * & n-p \end{pmatrix}. \tag{5}$$

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. *To generate observations, I sample Gaussian process* (1) *with*

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

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. Here are the results:*

	$\eta = 0.01$			$\eta = 0.05$		
	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$
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*

See here for source code.

Example 3.2. *For the next simulation, I modify the Gaussian process in Example 3.1 to include additional regressors*

$$\begin{aligned}\boldsymbol{x}((u,v)) &= \left(1, u, v, u^2, uv, v^2\right)' \\ \boldsymbol{\beta} &= (0.15, -0.65, -0.1, 0.9, -1.0, 1.2)'.\end{aligned}$$

Rerunning the simulation experiment with the same values of ℓ and η gives this result

	$\eta = 0.01$			$\eta = 0.05$		
	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 1.0$
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*

See here for source code.

To test prediction performance, we use Algorithm 2 which modifies Algorithm 1.

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

function PREDICTION-COVERAGE-TEST($N, \tilde{\boldsymbol{\theta}}, \alpha$)
  $cnt \leftarrow 0$
  for $i \leftarrow 1$ to N **do**
    $\tilde{\mathbf{y}} \leftarrow$ sample from $P(\mathbf{y} \mid \tilde{\boldsymbol{\theta}})$
    $t \leftarrow \int \mathrm{P}\left(\tilde{y}_1 \mid \boldsymbol{\theta}\right) \pi(\boldsymbol{\theta} \mid \tilde{y}_2, \ldots, \tilde{y}_n) d\boldsymbol{\theta}$
    if $\frac{\alpha}{2} < t < 1 - \frac{\alpha}{2}$ **then**
      $cnt \leftarrow cnt + 1$
    end if
  end for
  return $\frac{cnt}{N}$
end function

Example 3.3. *To generate observations, I sample from Gaussian process* (1) *with*

$$\sigma^2 = 1 \quad \psi_\ell(d) = \exp\left\{-\frac{d^2}{2\ell}\right\}.$$

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*. Below I show the coverage results for Bayesian prediction distributions using the reference prior and maximum likelihood prediction distributions:*

	$\eta = 0.001$			$\eta = 0.01$		
	$\ell = 0.1$	$\ell = 0.2$	$\ell = 0.5$	$\ell = 0.1$	$\ell = 0.2$	$\ell = 0.5$
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*

See here for source code.

4 Deterministic Bayesian Inference

The key component for deterministic inference is an accurate approximation to the posterior distribution 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

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

and $\boldsymbol{\Sigma}(\cdot)$ is defined in (5).

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

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

and for prediction distributions

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

Outline of Algorithm

Assume $\varphi_\ell(\cdot)$ and $\varphi_\eta(\cdot)$ are monotonically increasing surjective functions onto $(0,\infty)$. Put

$$\begin{aligned}
f(\mathbf{u}) = \; & -\log L^I(\varphi_\ell(u_1),\varphi_\eta(u_2);\mathbf{y}) \\
& -\log \pi\left(\varphi_\ell(u_1),\varphi_\eta(u_2)\right) - \log \dot{\varphi}_\ell(u_1) - \log \dot{\varphi}_\eta(u_2).
\end{aligned} \tag{6}$$

f is the negative of the reparameterized log of the posterior $\pi\left(\ell,\eta \mid \mathbf{y}\right)$. An approximation of f naturally leads to approximation and efficient integration of $\pi\left(\ell,\eta \mid \mathbf{y}\right)$.

We build an approximation $\tilde{f}$ in four steps

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

2. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ denote the two eigenvectors of the hessian at $\mathbf{u}_{\text{map}}$

   $$\nabla^2 f(\mathbf{u}_{\text{map}}).$$

   We find values $a_1 < 0 < b_1$, and $a_2 < 0 < b_2$ such that

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

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

3. We find monotonic cubic splines $s_1(\cdot)$ and $s_2(\cdot)$ such that

$$s_i(0) = a_i, \quad s_i(0.5) = 0, \text{ and } \quad s_i(1) = b_i$$

for $i = 1,\ 2$.

4. Put

$$g(\boldsymbol{x}) = f\left((s_1(x_1), s_2(x_2))'\right) - f(\mathbf{u}_{\text{map}}). \tag{7}$$

Using Chebyshev polynomials and the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ for a basis, we adaptively build a sparse polynomial to approximate g (and hence f) 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

$$\phi_\ell(t) = \phi_\eta(t) = \exp(t),$$

and 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 line search algorithm (e.g. Newton's method) for Step 2; we use the monotonic cubic algorithm from Fritsch, Carlson (1980) for Step 3. The algorithms for Step 1 and Step 4 are more complicated, and I break 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)

$$\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + \boldsymbol{s}_k \\
\boldsymbol{s}_k &= \underset{\boldsymbol{s}}{\operatorname{argmin}} \left\{ \nabla f(\boldsymbol{x}_{k-1})'\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}'\nabla^2 f(\boldsymbol{x}_{k-1})\boldsymbol{s} \right\} \\
&\qquad \text{s.t.} \quad \|\boldsymbol{s}\| \leq \delta_k.
\end{aligned}$$

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 3 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 (6).

Algorithm 3 Minimize Objective Function f

function MINIMIZE($f, \boldsymbol{x}_0$)
  $tol \leftarrow$ tolerance
  $\delta_0 \leftarrow$ an initial trust-region radius
  $v_0 \leftarrow f(\boldsymbol{x}_0)$
  $\boldsymbol{g}_0 \leftarrow \nabla f(\boldsymbol{x}_0)$
  $\boldsymbol{H}_0 \leftarrow \nabla^2 f(\boldsymbol{x}_0)$
  $k \leftarrow 0$
  while $\|\boldsymbol{g}_k\|_\infty > tol$ or not IS-POSITIVE-DEFINITE($\boldsymbol{H}_k$) **do**
    $\boldsymbol{x}_{k+1}, v_{k+1}, \delta_{k+1} \leftarrow$ COMPUTE-NEXT-STEP($\boldsymbol{x}_k, v_k, \boldsymbol{g}_k, \boldsymbol{H}_k, \delta_k$)
    $\boldsymbol{g}_{k+1} \leftarrow \nabla f(\boldsymbol{x}_{k+1})$
    $\boldsymbol{H}_{k+1} \leftarrow \nabla^2 f(\boldsymbol{x}_{k+1})$
    $k \leftarrow k + 1$
  end while
  return $\boldsymbol{x}_k, v_k, \boldsymbol{H}_k$
end function
function COMPUTE-NEXT-STEP($\boldsymbol{x}_k$, v_k, $\boldsymbol{g}_k$, $\boldsymbol{H}_k$, δ_k)
  $\delta_{k+1} \leftarrow \delta_k$
  while 1 **do**
    $\boldsymbol{s}_k \leftarrow \operatorname{argmin}_{\boldsymbol{s}} \left\{ \boldsymbol{g}_k' \boldsymbol{s} + \frac{1}{2} \boldsymbol{s}' \boldsymbol{H}_k \boldsymbol{s} \right\}$ s.t. $\|\boldsymbol{s}\| \leq \delta_{k+1}$
    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{s}_k$
    $v_{k+1} \leftarrow f(\boldsymbol{x}_{k+1})$
    $\rho \leftarrow \frac{v_{k+1} - v_k}{\boldsymbol{g}_k' \boldsymbol{s}_k + \frac{1}{2} \boldsymbol{s}_k' \boldsymbol{H}_k \boldsymbol{s}_k}$
    if $\rho < \frac{1}{4}$ **then**
      $\delta_{k+1} \leftarrow \frac{1}{4} \delta_{k+1}$
    else if $\rho > \frac{3}{4}$ and $\|\boldsymbol{s}_k\| = \delta_{k+1}$ **then**
      $\delta_{k+1} \leftarrow 2\delta_{k+1}$
    end if
    if $\rho > \frac{1}{4}$ **then**
      return $\boldsymbol{x}_{k+1}, v_{k+1}, \delta_{k+1}$
    end if
  end while
end function

Step 4: Sparse Polynomial Approximation

We seek to approximate $g(\cdot)$ (7) 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; but still, interpolating on a dense grid would be expensive. We can get better efficiency if we interpolate on a sparse grid, and we can get better efficiency still if we adaptively construct the sparse grid to avoid unnecessary evaluations in areas that can be approximated well by lower order polynomials.

Put

$$
\begin{aligned}
X^i &= \left\{x^i_1, \ldots, x^i_{m_i}\right\}, \\
m_i &= \begin{cases} 1 & \text{if } i = 0, \\ 2^{i-1} & \text{otherwise,} \end{cases} \\
x^i_j &= \begin{cases} 1 & \text{if } i = 0, \\ \frac{1}{2}\left(1 - \cos\frac{\pi(j-1)}{m_i - 1}\right) & \text{otherwise.} \end{cases}
\end{aligned}
$$

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 on sparse grids (Barthelmann et al., 2000; Klimke, 2006). Let $\psi^i_j(\cdot)$ denote the unique $m_i - 1$ degree polynomial where

$$\psi^i_j(x^i_{j'}) = \begin{cases} 1 & \text{if } j = j', \\ 0 & \text{otherwise,} \end{cases}$$

let V^i denote the vector space spanned by the basis functions $\left\{\psi^i_j\right\}$ for $j = 1, \ldots, m_i$, and define

$$
\begin{aligned}
\Delta V^0 &= V^0, \\
\Delta V^i &= V^i - V^{i-1} \quad \text{for } i > 0.
\end{aligned}
$$

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 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 $\mathcal{I}$, we follow Jakeman, Roberts (2011) and greedily add indexes and nodes with the largest approximation errors until a given accuracy is achieved. The algorithm adapts by both dimension and locality.

Algorithm 4 Approximate Function f over $[0,1]^d$

function APPROXIMATE(f)
  $tol \leftarrow$ tolerance
  $\mathcal{I} \leftarrow ()$
  $Z \leftarrow ()$
  $\boldsymbol{i} \leftarrow (0, \ldots, 0)$
  EVALUATE-SUBGRID(f, $\mathcal{I}$, Z, $\boldsymbol{i}$)
  $\mathcal{A} \leftarrow \{\boldsymbol{i}\}$
  while 1 **do**
    $\boldsymbol{i} \leftarrow \operatorname{argmax}_{\boldsymbol{i} \in \mathcal{A}} maxerr^{\boldsymbol{i}}$
    if $maxerr^{\boldsymbol{i}} < tol$ **then**
      return $\mathcal{I}, Z$
    end if
    $\mathcal{I} \leftarrow$ APPEND($\mathcal{I}, \boldsymbol{i}$)
    $Z \leftarrow$ APPEND($Z, \boldsymbol{z}^{\boldsymbol{i}}$)
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\boldsymbol{i}\}$
    for $\boldsymbol{i}_{\text{fwd}} \in \{\boldsymbol{i} + \boldsymbol{e}_k \mid 1 \leq k \leq d\}$ **do**
      if $\forall_k, (\boldsymbol{i}_{\text{fwd}})_k = 0 \vee \boldsymbol{i}_{\text{fwd}} - \boldsymbol{e}_k \in \mathcal{I}$ **then**
        EVALUATE-SUBGRID(f, $\mathcal{I}$, Z, $\boldsymbol{i}_{\text{fwd}}$)
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{i}_{\text{fwd}}\}$
      end if
    end for
  end while
end function
function EVALUATE-SUBGRID(f, $\mathcal{I}$, Z, $\boldsymbol{i}$)
  $maxerr^{\boldsymbol{i}} \leftarrow 0$
  $\boldsymbol{z}^{\boldsymbol{i}} \leftarrow \boldsymbol{0}$
  for $\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}} \in \Delta V^{i_1} \otimes \cdots \otimes \Delta V^{i_d}$ **do**
    if IS-ACTIVE($\boldsymbol{i}, \boldsymbol{j}$) **then**
      $y \leftarrow f(\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}})$
      $\tilde{y} \leftarrow$ EVALUATE($\mathcal{I}, Z, \boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{i}}$)
      $z_{\boldsymbol{j}}^{\boldsymbol{i}} \leftarrow y - \tilde{y}$
      $err_{\boldsymbol{j}}^{\boldsymbol{i}} \leftarrow |\exp(y) - \exp(\tilde{y})|$
      $maxerr^{\boldsymbol{i}} \leftarrow \max(maxerr^{\boldsymbol{i}}, err_{\boldsymbol{j}}^{\boldsymbol{i}})$
    end if
  end for
end function

function EVALUATE($\mathcal{I}$, Z, $\boldsymbol{x}$)
    $res \leftarrow 0$
    for $\boldsymbol{i}, \boldsymbol{z}^{\boldsymbol{i}} \in \mathcal{I}, Z$ **do**
        $res \leftarrow res + \sum_{z^{\boldsymbol{i}}_{\boldsymbol{j}} \in \boldsymbol{z}^{\boldsymbol{i}}} z^{\boldsymbol{i}}_{\boldsymbol{j}} \psi^{i_1}_{j_1}(x_1) \cdots \psi^{i_d}_{j_d}(x_d)$
    end for
    return res
end function
function IS-ACTIVE($\boldsymbol{i}$, $\boldsymbol{j}$)
    $\tau \leftarrow$ cutoff threshold
    for $k \in \{k \mid i_k > 0\}$ **do**
        $\boldsymbol{i}_{\text{bwd}} \leftarrow \boldsymbol{i} - \boldsymbol{e}_k$
        for $z^{\boldsymbol{i}_{\text{bwd}}}_{\boldsymbol{j}_{\text{bwd}}} \in \boldsymbol{z}^{\boldsymbol{i}_{\text{bwd}}}$ **do**
            if IS-POINT-NEIGHBOR($\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{j}_{\text{bwd}}$, k) **then**
                if $z^{\boldsymbol{i}_{\text{bwd}}}_{\boldsymbol{j}_{\text{bwd}}} > 0$ and $err^{\boldsymbol{i}_{\text{bwd}}}_{\boldsymbol{j}_{\text{bwd}}} > \tau$ **then**
                    return 1
                end if
            end if
        end for
    end for
    return 0
end function
function IS-POINT-NEIGHBOR($\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{j}'$, k)
    if $\exists_{k' \neq k}, j_{k'} \neq j'_{k'}$ **then**
        return 0
    end if
    if $i_k = 1$ **then**
        return 1
    end if
    if $j'_k > 1$ and $x^{i_k - 1}_{j'_k - 1} < x^{i_k}_{j_k} < x^{i_k - 1}_{j'_k}$ **then**
        return 1
    end if
    if $j'_k < m_{i_k - 1}$ and $x^{i_k - 1}_{j'_k} < x^{i_k}_{j_k} < x^{i_k - 1}_{j'_k + 1}$ **then**
        return 1
    end if
    return 0
end function

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

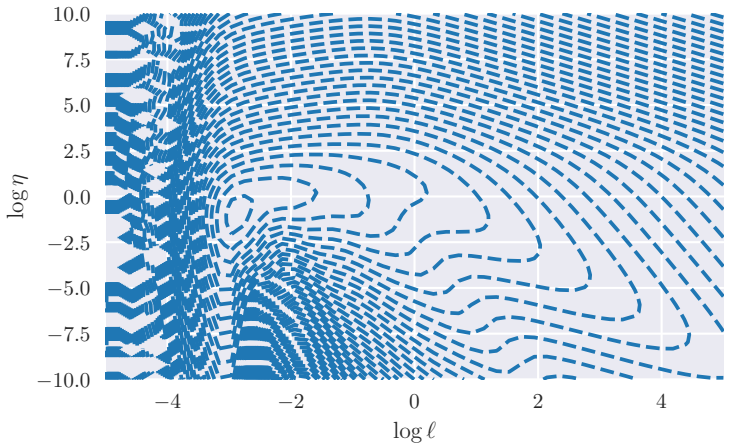

Figure 4: Reparameterized log posterior of the Example 1.1 data set

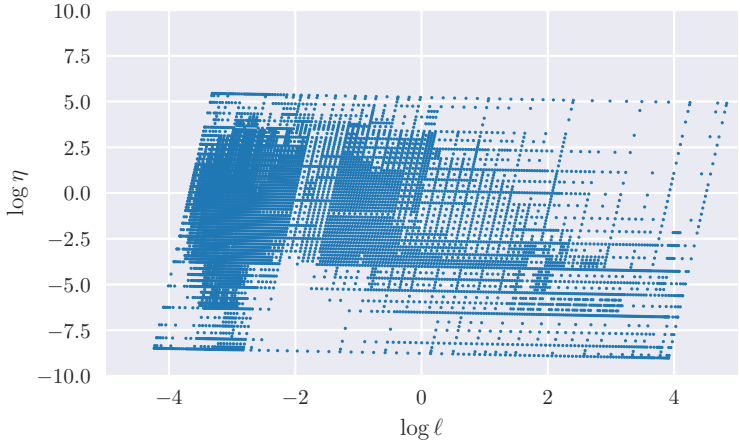

Figure 5: Sparse grid used to interpolate the log posterior

Prediction Distributions

The sparse grid from Algorithm 4 naturally leads to a quadrature rule to approximate integration (Jakeman, Roberts, 2011)

$$\begin{aligned}\int_0^\infty \int_0^\infty & h(\ell,\eta)\,\pi\left(\ell,\eta \mid \mathbf{y}\right) d\ell d\eta \\ &\approx \int_0^\infty \int_0^\infty h(\ell,\eta)\,\tilde{\pi}\left(\ell,\eta \mid \mathbf{y}\right) d\ell d\eta \\ &\approx \frac{1}{Z}\int_0^1 \int_0^1 h(\varphi_\ell(s_1(x_1)),\varphi_\eta(s_2(x_2)))g(x_1,x_2)\dot{s}_1(x_1)\dot{s}_2(x_2)d\ell d\eta \\ &\approx \sum_k w_k h(\ell_k,\eta_k), \end{aligned} \tag{8}$$

where the points $\{(\ell_k,\eta_k)'\}$ are chosen to be the transformed points of the sparse grid and weights are derived from integrals of the basis functions,

$$\int_0^1 \psi_j^i(x)dx.$$

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 $\mathrm{P}^\pi\left(\cdot \mid \mathbf{y},\ell,\eta\right)$. Use $\mathbf{y}_1 = \mathbf{y}$ to denote the observations and suppose $\mathbf{y}_2$ are possible values for the unobserved locations $\tilde{s}_1,\ldots,\tilde{s}_m$. Applying (2), we have

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

where R is given by (3). Put

$$\boldsymbol{R} = \begin{pmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{12}' & \boldsymbol{R}_{22}\end{pmatrix}.$$

Then

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

where

$$\begin{aligned}\bar{\mathbf{y}}_2 &= -\boldsymbol{R}_{22}^{-1}\boldsymbol{R}_{12}'\mathbf{y}_1 \\ b &= \mathbf{y}_1'\boldsymbol{R}_{11}\mathbf{y}_1 - \bar{\mathbf{y}}_2'\boldsymbol{R}_{22}\bar{\mathbf{y}}_2.\end{aligned}$$

σ^2 Marginal

The marginal distribution of σ^2 is given by

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

where $\{w_k\}$, $\{\ell_k\}$, and $\{\eta_k\}$ are given by (8). From (2), we have

$$\begin{aligned}\mathrm{P}^{\pi}\left(\sigma^2 \mid \mathbf{y}, \ell, \eta\right) &\propto \int_{\mathbb{R}^p} L(\boldsymbol{\beta}, \sigma^2, \ell, \eta; \mathbf{y}) \, \pi\left(\boldsymbol{\beta}, \sigma^2 \mid \ell, \eta\right) d\boldsymbol{\beta} \\ &\propto \left(\sigma^2\right)^{-(n-p)/2} |\mathbf{G}|^{-1/2} |\mathbf{X}'\mathbf{G}^{-1}\mathbf{X}|^{-1/2} \exp\left\{-\frac{S^2}{2\sigma^2}\right\} \left(\frac{1}{\sigma^2}\right) \\ &\propto \left(\frac{1}{\sigma^2}\right)^{(n-p)/2+1} \exp\left\{-\frac{S^2}{2\sigma^2}\right\}. \qquad (9)\end{aligned}$$

(9) 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

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

where

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

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 as a t-distribution with $n-p$ degress 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 4 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 $\{t_k\}$ are chosen by the Gauss-Legendre quadrature rule for the interval defined by the bracket in Step 2. If we evaluate $\tilde{\pi}\left(u_1 \mid \mathbf{y}\right)$ at Chebyshev nodes across the range of u_1 in the bracket, then we obtain a polynomial that approximates $\pi\left(u_1 \mid \mathbf{y}\right)$. $\pi\left(u_2 \mid \mathbf{y}\right)$ can be similarly approximated by a Chebyshev polynomial.

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

5 Real Data Analysis

Let's try applying the algorithms from Section 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

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

We'll use the same model and data set with our deterministic algorithm. When we fit a sparse polynomial to 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 4, 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. See here for source code for the example.

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

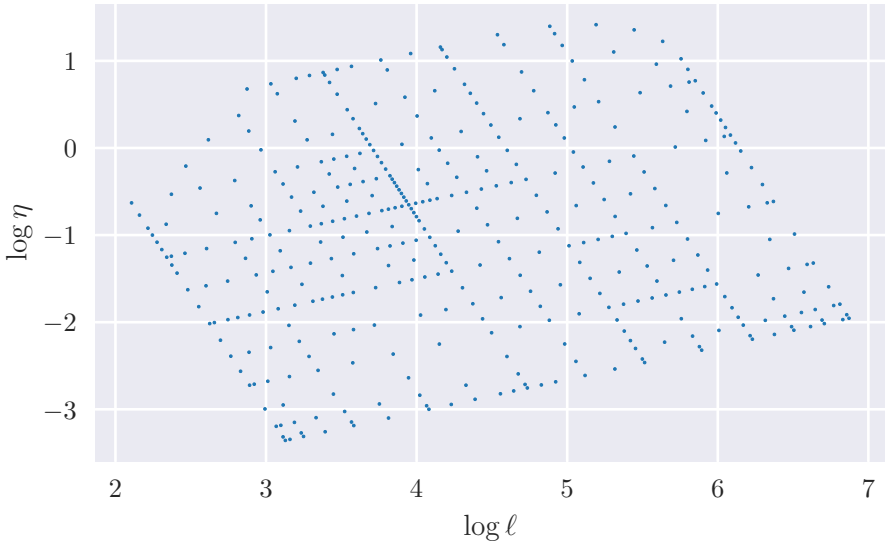

Figure 6: Sparse grid used to interpolate the posterior of the soil data set

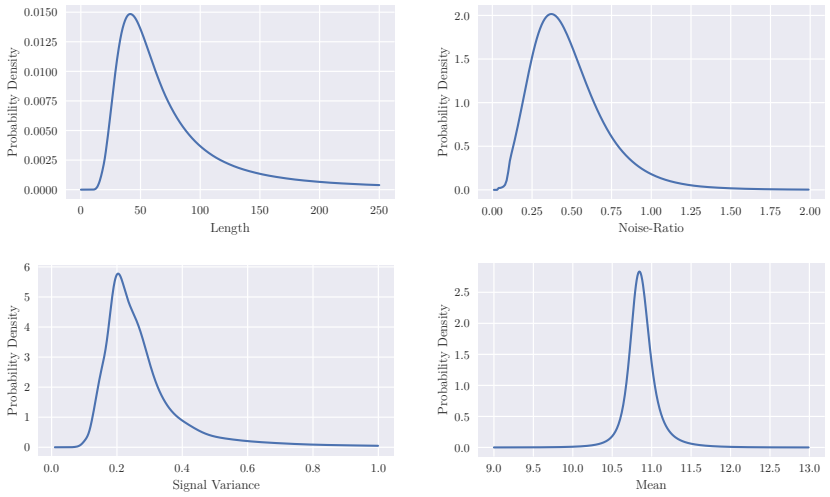

Figure 7: Marginalization of posterior distribution of soil carbon-to-nitrogen ratio data set

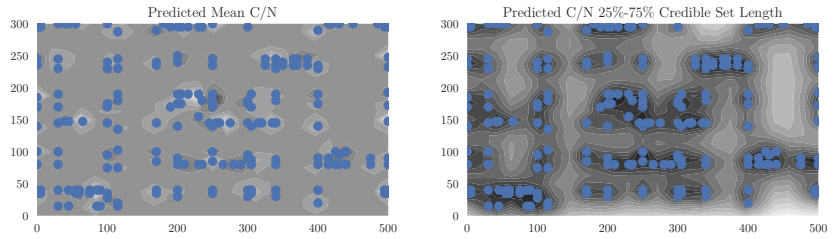

Figure 8: Prediction mean and credible sets of soil carbon-to-nitrogen ratios with sampling locations

(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 algorithm from §4. We model log Zinc concentration as a Gaussian process of the form (1) with

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

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 4, 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. See here for source code for the example.

A Appendix: Posterior Derivatives

We will derive equations to compute the value, gradient, and hessian of the negative log posterior $\pi\left(\ell, \eta \mid \mathbf{y}\right)$. From (2) and (4), we have

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

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 $\mathbf{G}$

$$\mathbf{G} = \boldsymbol{L}_G'\boldsymbol{L}_G.$$

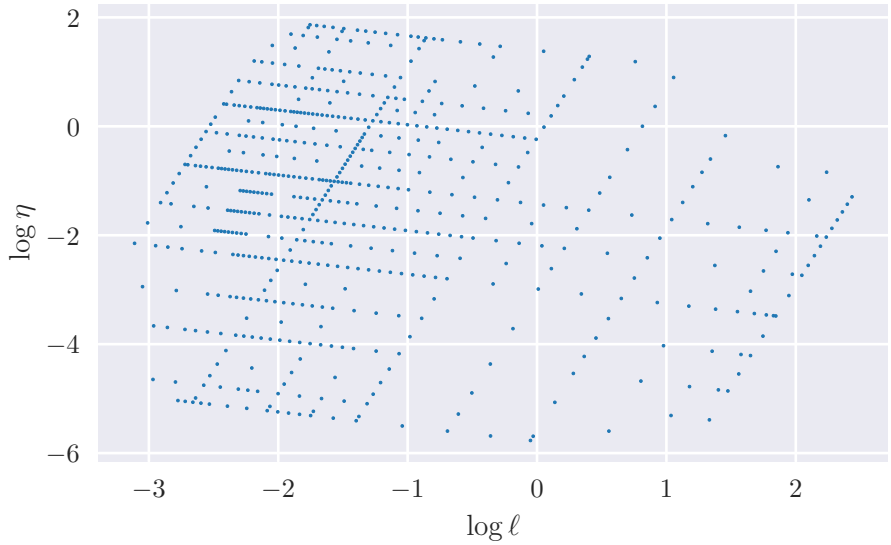

Figure 9: Sparse grid used to interpolate the posterior of the Meuse data set

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 $\boldsymbol{L}_G^{-1}\mathbf{X}$

$$\boldsymbol{A} = \mathbf{Q}\boldsymbol{R}_A.$$

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

$$\mathbf{H} = \mathbf{G}^{-1}\mathbf{X}\boldsymbol{A}^{-1}\mathbf{X}'\mathbf{G}^{-1} \quad \text{and} \quad \boldsymbol{F}_H = \boldsymbol{R}_A'^{-1}\mathbf{X}'\mathbf{G}^{-1}.$$

Applying to $\boldsymbol{R}$ (3), we have

$$\mathbf{H} = \boldsymbol{F}_H' \boldsymbol{F}_H \quad \text{and} \quad \boldsymbol{R} = \boldsymbol{L}_G'^{-1} \boldsymbol{L}_G^{-1} + \boldsymbol{F}_H' \boldsymbol{F}_H.$$

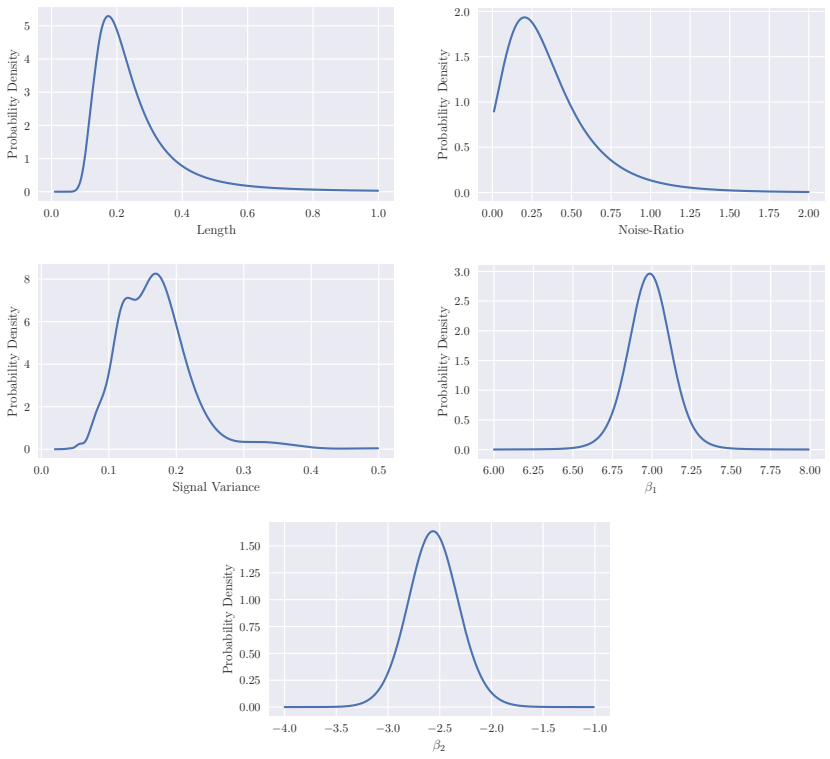

Figure 10: Marginalization of posterior distribution of Meuse data set

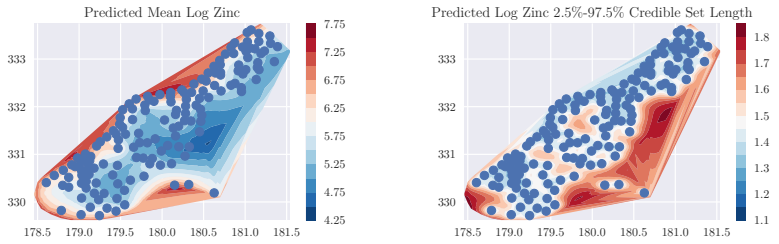

Figure 11: Prediction mean and credible sets of Meuse data set with sampling locations

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

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

Using this 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

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

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

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

Hessian

Computing second derivatives we have

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

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

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

and

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

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

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

and

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

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, 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.

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.

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.

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.