Chapter 3

Methodological Development

3.1 Introduction

In this chapter, I present the basic methodology for using stationary and nonstationary covariance models as components in a Bayesian hierarchical model. I open by presenting methods for parameterizing the Gaussian kernels of the nonstationary correlation functions that I present in Chapter 2. Numerical sensitivity is an important issue for Gaussian process (GP) models because of the high correlation between the function values and the resulting numerical singularity of covariance matrices. I give an overview of approaches for dealing with this. I describe parameterizations for GP models, discuss some of the issues involved in particular parameterizations, and present the parameterization I have chosen to use. One difficulty is that the GP model involves an inherent degree of non-identifiability that I will describe. Next I discuss Markov chain Monte Carlo (MCMC) proposal schemes, describing previous approaches and discussing the mixing difficulties involved in sampling GP models. These schemes deal with the numerically-singular matrices in different ways, and I describe the issues involved. I suggest a new type of proposal scheme, which I call posterior mean centering (PMC), as a way to improve mixing and provide evidence that mixing improves with this scheme when the process cannot be integrated out of the model. The PMC scheme is implemented so as to avoid problems with singularity. One of the major drawbacks to GP models, including the models discussed in this thesis, is the $O(n^3)$ computation involved in working with the covariance matrices; I close by giving an overview of some approaches that have

been suggested for speeding up the computations.

For clarity, in the sections that follow, with the exception of Section 3.2, I will assume the following simple model for the data and parameters:

$$Y_i \sim \mathrm{N}\left(f\left(x_i\right), \eta^2\right) \tag{3.1}$$

$$f(\cdot) \sim \mathrm{GP}\left(\mu, \sigma^2 R(\kappa)\right) \tag{3.2}$$

$$(\mu, \sigma, \kappa, \eta) \sim \Pi(\mu) \cdot \Pi(\sigma) \cdot \Pi(\kappa) \cdot \Pi(\eta), \tag{3.3}$$

with a stationary correlation function, $R(\kappa)$, parameterized by a scalar correlation scale parameter κ. In the nonstationary models considered in Chapters 4 and 5, κ is replaced by the convolution kernels and attendant hyperparameters used to construct the nonstationary correlation function. The methods and discussion given here apply to those more complicated hierarchical models with embedded Gaussian process distributions. Here and elsewhere in this thesis, as necessary, I take $\mu = \mu\mathbf{1}$, when a vector-valued mean is required.

3.2 Parameterizing the Kernels for Nonstationary Covariance Models

In Higdon et al. (1999)'s convolution approach to constructing nonstationary covariance functions, the kernels completely determine the nonstationary covariance structure of the GP. In Section 2.5.5, I show that to retain the smoothness of the underlying stationary correlation function upon which a kernel-based nonstationary correlation function is constructed, it is required that the kernel matrices vary smoothly in space. Using Gaussian kernels, this means that we need positive definite matrices that vary smoothly. The second key challenge is that the prior be uniform over the orientations of the Gaussian kernels.

3.2.1 Two-dimensional foci approach

Higdon et al. (1999) parameterize spatially-varying positive definite matrices based on the foci of the one standard deviation ellipse of the Gaussian distribution on $\Re^2$. Gaussian process priors with stationary squared exponential covariance functions are placed on the x and y coordinates

of one focus. They force the areas of the ellipses for different locations to be the same, but not fixed, value. Swall (1999, p. 94) allows this area to vary but finds that the model overfits when sampling the variance and correlation scale parameters for the GP determining the area; she fixes these hyperparameters and suggests reparameterization or more informative prior distributions to solve the problem. It is not clear why this overfitting occurred; I have generally not found this to be a problem with the parameterizations used in the regression modelling in this work. Hilbert and Cohn-Vassen (1952) describe a generalization of the focal construction of an ellipse to ellipsoids in three dimensions. However the construction is complicated and extensions to higher dimensions are not apparent. I have not found other means of intuitively constructing high-dimensional ellipsoids. Hence, while the focal approach is feasible for the storm data of Chapter 5, other approaches are needed for data in higher dimensional spaces, such as the regression modelling of Chapter 4. For both the regression modelling and spatial modelling, I choose to use the eigendecomposition approach described in Section 3.2.3.

3.2.2 Cholesky decomposition

One approach to modelling smoothly-varying positive definite matrices in higher dimensions uses the Cholesky decomposition of the matrix, $\Sigma = LL^T$, where L is lower triangular. If one chooses the square of the diagonal elements, $L_{p,p}^2 \sim \chi_{d-p+1}^2, d > P-1, p = 1, \ldots, P$, and the off-diagonals to be $\mathrm{N}(0,1)$, with all the elements independent, then Σ will be positive definite and distributed Wishart (d, I) (Odell and Feiveson 1966). Now consider matrices, $\Sigma_{\boldsymbol{x}}$, defined at every point in the space. To produce spatially-varying positive definite matrices, let each off-diagonal element, $L_{i,j}(\cdot)$, be a random process distributed spatially according to Gaussian process with mean zero and a correlation function as the covariance function, thereby giving standard normal marginal distributions at each location. Let the GPs for each of the elements be independent. For the diagonal elements, $L_{p,p}(\cdot)$, use independent Gaussian processes with correlation function as the covariance function and have the diagonal elements be the square root of the inverse χ_{d-p+1}^2 CDF transformation of the Gaussian process realizations. The result is spatially varying positive definite matrices that are marginally Wishart (d, I), and that vary smoothly in space according the correlation matrices specified for the underlying GPs of the elements of the matrices. Since the

scale matrix of the constructed Wishart marginal distribution is the identity matrix, the marginal distribution for the resulting positive definite matrix at each location is rotationally symmetric.

Unfortunately, with this approach the marginal distributions of the ratios of the eigenvalues at a location change in concert with d, the value determining the degrees of freedom of the χ^2 random variables. The larger is d, the smaller is the magnitude of the larger eigenvalues relative to the smaller eigenvalues. In other words, kernels that spread over a large area tend to be spherical in shape. A further difficulty is that since the variances are marginally χ^2_d, it is difficult to express lack of prior information about the size of the variances since a large value for d results in little prior weight on small variances. Because of this difficulty in jointly specifying the prior variance and eigenvalue ratios, I choose instead to use the eigendecompositions of the kernel matrices directly.

3.2.3 Eigendecomposition

The eigendecomposition of a positive definite matrix is $\Sigma = \Gamma\Lambda\Gamma^T$, where Λ is a diagonal matrix of positive eigenvalues, and Γ is a matrix of eigenvectors, an orthonormal matrix.

3.2.3.1 Givens angles

A straightforward, and minimally parameterized, specification of a positive definite matrix is through its eigenvalues and the Givens angles used to construct its eigenvector matrix, which is a rotation matrix. Anderson, Olkin, and Underhill (1987) show that any orthogonal matrix can be expressed as the product of Givens matrices, G_{ij}, and a matrix, D_ϵ, of reflections:

$$\Gamma = (G_{12}G_{13}\cdots G_{1P})(G_{23}\cdots G_{2P})\cdots(G_{P-1,P})D_\epsilon \ ,$$

where

$$G_{ij} = G_{ij}(\rho_{ij}) = \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} \quad & i & \quad & j & \quad \end{array} \\ \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & \cos\rho_{ij} & 0 & -\sin\rho_{ij} & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & \sin\rho_{ij} & 0 & \cos\rho_{ij} & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \end{array}$$

and $\rho_{ij} \in (-\frac{\pi}{2}, \frac{\pi}{2})$ is the angle of rotation in the i, j plane. Anderson et al. (1987) further show how to produce a random orthogonal matrix with Haar measure over the orthogonal group, where the distribution over Γ is the same as the distribution over $\Upsilon\Gamma$ for all orthogonal Υ. Take the elements of D_ϵ to be ± 1 independently with probability $\frac{1}{2}$. Let ρ_{ij} be independent with density proportional to:

$$\left(\prod_{j=2}^{P} \cos^{j-2} \rho_{1j}\right)\left(\prod_{j=3}^{P} \cos^{j-3} \rho_{2j}\right)\cdots\left(\prod_{j=P}^{P} \cos^{j-P} \rho_{P-1,j}\right). \tag{3.4}$$

If the marginal prior distribution for a single matrix Γ is specified in this way, then Σ is rotationally invariant since Γ has Haar measure.

This approach includes the matrix of reflections, which cannot be parameterized to vary smoothly in space. Fortunately, this matrix can be omitted without changing the resulting kernel matrices. Each kernel matrix, Σ, can be represented as

$$\Sigma = (G_{12}\cdots G_{1P})(G_{23}\cdots G_{2P})\cdots(G_{P-1,P})D_\epsilon \Lambda D_\epsilon (G_{P-1,P})^T (G_{2P}^T \cdots G_{23}^T)(G_{1P}^T \cdots G_{12}^T),$$

and since $D_\epsilon \Lambda D_\epsilon = \Lambda$, we see that D_ϵ is unnecessary. The attractive features of this parameterization are that it is simple to specify prior distributions such that each positive definite matrix is uniform over rotations and that the minimum number of parameters, $P + \frac{P(P-1)}{2}$ are used.

The primary drawback to this approach is the difficulty of parameterizing angles, ρ_{ij}, that vary smoothly in space. One can model non-normal random variables whose domain is monotonic as stochastic processes using the appropriate inverse CDF transformation of an underlying Gaussian process, but the lack of monotonicity of angular-valued random variables prevents that approach here. In the next section, I present an overparameterized alternative that works around this problem.

There are cases in which I need to parameterize individual correlation matrices and am not concerned with the spatial correlation structure of multiple matrices; in these cases I use the eigendecomposition with the Givens angle parameterization of the eigenvector matrix given above. To facilitate mixing, at the expense of identifiability, I expand the support of ρ_{ij} to $(-\pi, \pi)$. This allows smooth movements around the parameter space, without artificial boundaries at $\rho_{ij} = \pm\frac{\pi}{2}$. Furthermore, when a value of ρ_{ij} is proposed outside of $(-\pi, \pi)$, I replace it with $\rho_{ij} + c2\pi$, which gives the same eigenvector matrix, but keeps ρ_{ij} in its support when the integer c is chosen appropriately. The new prior is the same as before (3.4), but with $|\cos \rho_{ij}|$ in place of $\cos \rho_{ij}$. This can

be seen to preserve the uniform prior over rotations because (3.4) arises in the proof in Anderson et al. (1987) as the Jacobian of a transformation; for $\rho_{ij} \in (-\pi, -\frac{\pi}{2}) \bigcup (\frac{\pi}{2}, \pi)$, we need to use the absolute value of the Jacobian to produce the density. Alternatives to the Givens angle parameterization for individual correlation matrices are discussed in Lockwood (2001, p. 58), including the uniform distribution over correlation matrices.

3.2.3.2 Overparameterized eigenvectors

To avoid the difficulties in working with smoothly-varying angles, I overparameterize the eigenvector matrix. For now, let's consider a single eigenvector matrix at a location. Instead of working with $P + \frac{P(P-1)}{2}$ parameters per positive definite matrix, I work with $P + \frac{P(P-1)}{2} + P - 1$ parameters. I construct the first eigenvector from P independent random variables by normalizing the variables collected into a vector. Then, in each successively smaller-dimensional subspace, I construct eigenvectors by collecting successively one fewer random variable into a normalized vector. I construct the final orthogonal matrix as the product of the orthogonal matrices in the subspaces, imposing constraints as necessary.

An example in three dimensions will clarify the setup. I construct the eigenvector matrix, Γ, as $\Gamma = \Gamma_2\Gamma_1$. The elements of Γ_2 are determined from the realizations, x, y, z, of three random variables, X, Y, Z, in the following fashion

$$\Gamma_2 = \begin{pmatrix} \frac{x}{l_{xyz}} & \frac{-y}{l_{xy}} & \frac{-xz}{l_{xy}l_{xyz}} \\ \frac{y}{l_{xyz}} & \frac{x}{l_{xy}} & \frac{-yz}{l_{xy}l_{xyz}} \\ \frac{z}{l_{xyz}} & 0 & \frac{l_{xy}}{l_{xyz}} \end{pmatrix},$$

where $l_{xyz} = \sqrt{x^2 + y^2 + z^2}$ and $l_{xy} = \sqrt{x^2 + y^2}$. In this setup $(\Gamma_2)_{32} = 0$ is the required additional constraint on Γ_2, and the remaining elements of the matrix are fully determined. (Note that if the first eigenvector is extremely close to $(0, 0, 1)$, I need an additional constraint, so I also set $(\Gamma_2)_{22} = 0$.) I now descend to the two dimensional subspace orthogonal to Γ_2. Let

$$\Gamma_1 = \begin{pmatrix} \frac{u}{l_{uv}} & -\frac{v}{l_{uv}} \\ \frac{v}{l_{uv}} & \frac{u}{l_{uv}} \end{pmatrix}.$$

Here, the elements of the matrix are fully determined by two random variables, U and V, using a

simplification of the construction of Γ_2 above. The final eigenvector matrix is the product, $\Gamma = \Gamma_2\Gamma_1$.

To ensure that the eigenvector matrices and hence the kernel matrices vary smoothly in space, I place GP priors, with mean 0, variance 1 and a correlation function, on the random variables used to construct the eigenvectors (in three dimensions, these are X, Y, Z, U and V). In doing this conversion from angular space to Euclidean space, I have used one extra parameter for each dimension (save the last), relative to the minimally-parameterized Givens angle approach. The kernel matrix parameterization is completed by taking Gaussian process priors for the log of each of the eigenvalue processes, with the mean and variance as hyperparameters. Note that this construction achieves uniformity over rotations, since the eigenvectors in the subspaces are uniformly distributed over rotations. It also achieves smoothness in space, provided that X, Y, Z, U, and V vary smoothly. Note that the elements of Γ_2 and Γ_1, and hence Γ as well, are infinitely differentiable as functions of X, Y, Z, U, and V, so the elements of the kernel matrices will be differentiable to the same degree as the stochastic processes used to construct the elements. If I take X, Y, Z, U, and V to be highly differentiable stochastic processes, then using the result in Section 2.5.5, the smoothness of nonstationary stochastic processes with kernel convolution covariance will depend on the smoothness properties of the underlying stationary correlation, which is my goal.

The parameterization in two dimensions involves only Γ_1. In higher dimensions, it is possible to parameterize the eigenvector matrix in similar fashion to that described above for three dimensions, although working out the correct constraints for Γ_3 and larger matrices is tedious, and it remains an open question as to whether a model with so many processes for the eigenvectors will mix well. There may be representations of the angles/eigenvectors that are simpler than that proposed above.

3.2.3.3 Simplified eigendecomposition model

If one were to use the construction just described to model nonstationary covariance in P-dimensional covariate spaces, one would need $P + \frac{P(P-1)}{2} + P - 1$ processes, with that many prior correlation functions. If these prior correlation functions were, for example, fully anisotropic stationary squared exponential correlations, each would have a matrix Δ in the Mahalanobis distance calcula-

tion in its correlation function, $R(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(-\frac{1}{2}\left(\boldsymbol{x_i} - \boldsymbol{x_j}\right)^T \Delta^{-1}\left(\boldsymbol{x_i} - \boldsymbol{x_j}\right)\right)$, which would require $P + \frac{P(P-1)}{2}$ parameters, making for $\left(2P - 1 + \frac{P(P-1)}{2}\right) \cdot \left(P + \frac{P(P-1)}{2}\right)$ parameters just for the correlation structure of the eigenprocesses. Even in three dimensions, this starts to be unwieldy, with 48 parameters, and could lead to overfitting, so we might think about using simpler prior structures for the eigenvalues and eigenvectors. One possibility, which I employ in the regression modelling, builds on the observation that the primary goal is to have the kernels vary smoothly, and the eigenprocesses are only a means to that end. Instead of having $2P - 1 + \frac{P(P-1)}{2}$ different prior correlation functions, use the same prior correlation function for all the eigenvalue and eigenvector processes. This says that all of the processes used to construct the kernels have the same correlation structure. Note that the different eigenvalues will still have their own individual mean and variance hyperparameters, so that the scales of the eigenvalues may differ.

Other possibilities include requiring that the kernels be axis-oriented, so that the eigenvectors are not modelled and there are only P eigenvalue processes to model, having the eigenvector matrix be the same for all locations so that we need only $\frac{P(P-1)}{2}$ Givens angles to parameterize the eigenvector matrix, or having the eigenprocesses have simple one-parameter correlation functions (i.e., $\Delta = \frac{1}{\kappa} I$).

3.2.4 Basis kernel model

Modelling the eigenvectors and eigenvalues as Gaussian processes requires sampling Gaussian processes in the hierarchy of the model. In addition to the general difficulties involved in sampling from GPs described later in this chapter, when the GPs are not directly involved in the likelihood, the sampling may be particularly difficult. For problems in which the correlation scale of the data is unknown, such as devising a generic regression methodology, using GPs for the kernels may be the best approach. However, for problems in which the approximate correlation scales are known in advance, it may be possible to specify a much simpler model for the kernel matrices that is easier to fit, in particular, easier to sample from via MCMC. Higdon (1998) proposed a basis kernel approach in which a small number of basis kernels are parameterized, and kernels at any location of interest are weighted averages of the basis kernels using a simply-parameterized weight decay function. This reduces the number of parameters required to specify the nonstationary

covariance structure and allows for easier sampling because one does not need to deal with the covariance structure of the kernels except via a parameter that determines the degree of locality in the weighted averaging of the basis kernels. The basis kernels should be located closely enough together that they can capture the nonstationary in the data, but far enough apart that they can be assumed independent a priori and so that there are relatively few of them. Since we do not need to explicitly specify correlation between the basis kernels, we can use the Givens angle approach to minimally parameterize the positive definite basis kernel matrices, giving P parameters for the eigenvalues and $\frac{P(P-1)}{2}$ parameters for the angles of each covariance matrix. In modelling the storm data in Chapter 5, I use nine Gaussian basis kernels spread over one third of the Northern Hemisphere, giving me $9 \cdot \left(P + \frac{P(P-1)}{2}\right) = 27$ parameters for the correlation structure. Higdon (1998) modelled ocean temperature data in a portion of the Atlantic Ocean using a set of 8 basis kernels. In practice the basis kernels appear to pick up some nonstationary features of the data, and the basis kernel and weight parameters seem to mix adequately. For the weight decay function, I use the squared exponential function (i.e., a Gaussian density function), but it does not have to be a positive definite function. However note that sample path smoothness will depend on whether the kernel matrices produced by weighted averaging of the basis kernels are sufficiently smooth, so the form of the function does matter. As the dimensionality of the space increases, more basis kernels are needed to cover the space to the same degree of density, so this approach is probably not feasible in high dimensions.

3.3 Numerical Sensitivity of GP Models

One difficulty in using GP models involves the numerical calculation of matrix square roots, solutions of systems of linear equations, and, depending on the fitting methods, matrix inverses. In particular, as the correlation scale increases, correlation matrices can approach numerical singularity. The smallest eigenvalues get so close to zero that the limitations of finite-precision arithmetic come into play, and numerically negative eigenvalues occur. This particularly affects the squared exponential correlation function, but also occurs with the Matérn even for relatively small values of ν. For example, using the R statistical software, which I believe has a relatively accurate Bessel function, and generating 100 points randomly on (0,1), the correlation matrix for the points was

numerically singular with $\kappa = 0.25$ and $\nu = 4$ in 7 of 10 draws. Note that with $\nu = 4$ the sample paths are not quite three times differentiable. For $\kappa = 1.0$ and $\nu = 2$, in 7 of 100 draws the correlation matrix was singular; here the sample paths are not quite twice differentiable. The problem is acute with the Matérn and squared exponential correlation functions, because they specify that at small distances, locations are very highly correlated, with the derivative of the correlation function at 0 being 0, whereas the derivative of the exponential correlation function at 0 is negative.

To understand the problem in detail, let $\boldsymbol{f} = L\boldsymbol{\omega}$ where $LL^T = C = \text{Cov}(\boldsymbol{f})$ and $\boldsymbol{\omega} \sim \text{N}(0, I)$. The specific limitation can be seen in the calculation of the lower triangular square root matrix, or Cholesky factor, L, of C. One approach to calculating the Cholesky proceeds column by column. Consider the calculated diagonal element for the ith column, $L_{i,i} = \sqrt{C_{i,i} - \sum_{j=1}^{i-1} L_{i,j}^2}$, which can be interpreted as the residual variance of f_i, the ith element of $\boldsymbol{f}$, after regressing on $f_1, \ldots, f_{i-1}$. When f_i is very highly correlated with $f_1, \ldots, f_{i-1}$, $\sum L_{i,j}^2$ can be affected by round-off error in the calculations and $L_{i,i}^2$ can be smaller than machine precision (usually O $\left(10^{-16}\right)$ for double precision floating point values). When this happens, it means that the f_i is nearly a linear combination of $f_1, \ldots, f_{i-1}$. Several solutions have been proposed for this problem.

A straightforward approach when one only needs the Cholesky and not the inverse of the Cholesky, is to acknowledge that f_i is essentially a linear combination of $f_1, \ldots, f_{i-1}$ by setting $L_{i,i} = 0$ and doing the same for all the elements of that column of the Cholesky, $L_{i+1,i}, \ldots, L_{n,i}$ (Lockwood et al. 2001). In practice one sets a threshold, ϵ, and if the calculated value, $L_{i,i}^2 < \epsilon$, then the column is zeroed out. This states that the random variable is exactly a linear combination of the other random variables. In doing so, one makes the process smoother than it actually is, since one ignores the small amount of residual variability in the random variable. Using a different approach, Swall (1999) solved this problem in an elegant fashion by adaptively reordering the columns of the matrix so that the diagonal elements of L decreased monotonically. Upon reaching a predetermined tolerance level, she declared the remaining elements of $\boldsymbol{f}$ to be linear combinations of the previous elements. This adaptive approach seems likely to be more accurate than that of Lockwood et al. (2001). However, in part to minimize computation, I have employed the Lockwood et al. (2001) approach, which seems sufficiently accurate, provided the columns are not ordered such that many nearby locations with high correlation are closely grouped within the matrix (see below for more

details). While both these approaches allow one to calculate an approximation to the Cholesky, if one needs the inverse of the Cholesky to solve a set of linear equations, then the result is values of infinity in the inverse matrix, which prevent further calculation, such as finding $\boldsymbol{\omega} = L^{-1}\boldsymbol{f}$, which is needed to calculate the prior density of $\boldsymbol{f}$. If the ith column is zeroed out, the value of the ith element of $\boldsymbol{\omega}$, ω_i, is unknown since it is not used in calculating $\boldsymbol{f} = L\boldsymbol{\omega}$. One possibility if one needs the inverse would be to set $\omega_i \sim \mathrm{N}(0, 1)$, namely a random deviate from the prior on $\boldsymbol{\omega}$, although one would need to think through the implications of this before proceeding. In addition, with this generalized Cholesky algorithm, one cannot calculate the determinant of the matrix, which is also needed to calculate the prior for $\boldsymbol{f}$. Swall (1999) integrated the spatial mean process out of the model and only outside of the main Markov chain did she sample the process in a Gibbs step conditional on the hyperparameters, thereby avoiding the inverse and determinant calculations. However in sampling the spatial foci processes and their hyperparameters (Section 3.2.1), it is not clear how she avoided inverse and determinant calculations with ill-conditioned matrices, since these Metropolis steps would have required calculation of the GP prior for the focal processes.

In the models used in chapters 4 and 5, I set the tolerance of the generalized Cholesky algorithm to either $\epsilon = 10^{-10}$ or $\epsilon = 10^{-12}$. This was based on calculations with various tolerances and parameter values and with distances consistent with those used in the regression and spatial models. In the calculations, I reconstructed $C' = LL^T$ and compared the approximation, C', to the original C, although it is not clear which loss function to use in comparing the original and approximate covariance matrices. Setting the tolerance to a smaller value increases the error by keeping inaccurate values in L, while setting the tolerance to larger values increases error by zeroing out more columns. Based on some additional ad hoc experimentation, for future work, I would suggest a tolerance on the order of $\epsilon = 10^{-9}$.

One might think that if the matrix is numerically singular that calculating a generalized inverse would be a good approach. However, all the generalized inverse provides is one of multiple solutions to the system of linear equations. Also, it doesn't give you the determinant. It does not resolve the problem with numerical singularity, which is that we can't know what some of the elements of $\boldsymbol{a} = C^{-1}\boldsymbol{b}$ are. For the proposal schemes I describe in Section 3.6 one needs a square root of the

covariance matrix, so the generalized Cholesky algorithm described above is sufficient, and a generalized inverse is not. An alternative would be to use the eigendecomposition of C and set all the eigenvalues less than some tolerance to zero, with $C^{\frac{1}{2}} = \Gamma\Lambda^{\frac{1}{2}}$. I have not investigated whether this approach gives a more accurate approximation to C than does the generalized Cholesky, but this may be worth more assessment. In the parameterizations and MCMC sampling schemes described in Sections 3.4 and 3.6, one prominent issue will be the need or lack thereof to invert the Cholesky of the covariance. I will present a parameterization and sampling scheme that work with only with the generalized Cholesky and not its non-existent inverse or determinant.

One effect of this numerical challenge is that the correlation function needs to be extremely accurate numerically. This is particularly an issue in calculating the Bessel function at the heart of the Matérn correlation function. An example of the sensitivity is that in running the same code on two different Linux PCs, one with an Intel Pentium processor and the other with an AMD Athlon processor, a difference of O $\left(10^{-15}\right)$ in the calculation of a covariance matrix element resulted in eventually changing the quantitative (though not qualitative) results of an MCMC run. Another example is that during MCMC, I have found that for certain sets of parameter values, I am not able to calculate the covariance matrix sufficiently accurately, and the resulting 'Cholesky' decomposition does not in fact come very close to reconstructing the original covariance such that $LL^T \approx C$. This occurs infrequently, but the result when it happens and the proposal is accepted is an erroneous sample path with local jaggedness that is not consistent with high local correlations specified by the hyperparameters. To illustrate the problem, using the built-in Bessel function in the statistical software R (besselK), I generated 5000 Matérn correlation matrices using $\log\kappa \sim \text{U}(\log(.03), \log(2))$ and $\nu \sim \text{U}(0.5, 30)$ and random permutations of a vector of 100 values of $x \sim \text{U}(0, 1)$. For a small number of correlation matrices (on the order of several dozen), the resulting generalized Cholesky was not a close approximation to the true Cholesky factor. There was no clear pattern relating the values of κ and ν to the inaccuracy, although the worst cases seemed to occur with relatively low values of κ and with values of ν greater than 10. Setting the tolerance at a larger value ameliorates the problem in the cases in which it arises, but at the expense of less accurate generalized Cholesky factors for most other parameter settings. Slightly changing the values of κ and ν removes the problem for an individual matrix. In Figure 3.1 I show an example of a sample function produced

with an inaccurate generalized Cholesky factor.

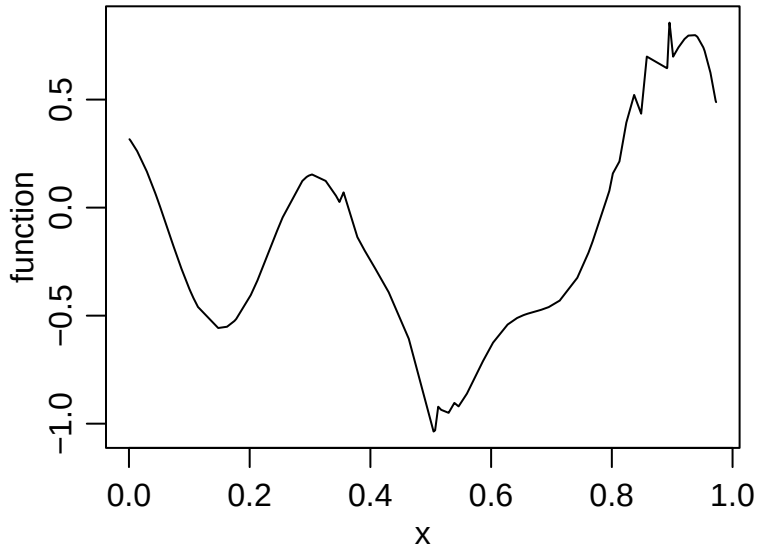

Figure 3.1. Sample function drawn from $f(\cdot) \sim GP(0, R(\kappa = 0.17, \nu = 25))$ *with an inaccurate generalized Cholesky factor. Note the jaggedness in the function at several points.*

In the machine learning literature, the standard solution to numerical singularity is to introduce a small amount of jitter on the diagonal by adding some small value δ to each diagonal element (Neal 1997). Provided the jitter is large enough, the matrix is no longer singular and can be inverted. If the jitter is not too large, then the result will hopefully be similar to the result that would be obtained under infinite-precision arithmetic. However, adding jitter raises some obvious questions of interpretation. A Gaussian process model specifies that nearby locations are highly correlated and that the resulting process is therefore smooth. Introducing jitter makes the process discontinuous, regardless of the correlation function used, because the covariance function after including the effect of jitter is not a continuous function at the origin. In practice the effect of jittering may not materially change the resulting function estimates, but this solution seems troubling. Machine learning researchers tend to set the mean of the GP to zero, which makes the covariance matrix even more ill-conditioned when the observations suggest that the realized function is far from zero, resulting in fitted correlations very close to one.

A final numerical issue involving the covariance matrix is the order of the locations in the

covariance matrix. Because the zeroing out of a column in the generalized Cholesky algorithm is dependent on the correlation of the location corresponding to that column with the locations whose columns came previously, the accuracy of the approximation depends on the ordering of the columns. When the correlations are high, one should avoid having many neighboring locations as the early columns in the covariance, as this results in zeroing out columns early in the Cholesky, which can drastically lower the accuracy of the approximation. In practice, I use a random permutation of the columns to avoid placing many nearby locations in nearby columns of the covariance matrix. Another sensitivity with respect to the Cholesky arises because of the key role played by the first column and its location. The values of the random process are all conditional on the value of the process at this first location, so the ordering can affect the mixing of the chain. In Figure 3.2, I show an example of this in which the order affects the mixing of the correlation scale parameter for the kernel eigenvalue process, but seemingly not the other hyperparameters, in a one-dimensional regression problem with $f(x) = \sin \frac{1}{x}$.

3.4 GP Parameterizations

3.4.1 Centered vs. noncentered parameterizations

There are two straightforward parameterizations of a Gaussian process. The first parameterization is simply that for a finite set of values, $\boldsymbol{f}$, from the process

$$\boldsymbol{f} \sim \mathrm{N}(\mu, \sigma^2 R(\kappa)).$$

In this parameterization, which I will call the 'centered' parameterization for reasons that will become apparent, the function $\boldsymbol{f}$ has hyperparameters $\mu, \sigma,$ and κ, which reside at a level one higher in the model hierarchy than does $\boldsymbol{f}$. In the second, 'uncentered', parameterization, we have

$$\boldsymbol{f} = \mu + \sigma L(\kappa)\boldsymbol{\omega} \tag{3.5}$$

$$\boldsymbol{\omega} \sim \mathrm{N}(0, I), \tag{3.6}$$

where $L(\kappa)$ is the Cholesky factor of $R(\kappa)$. Here, $\boldsymbol{f}$ is a deterministic function of the parameters $\boldsymbol{\theta} = \{\mu, \sigma, \kappa, \boldsymbol{\omega}\}$, which together reside at the same level of the model hierarchy.

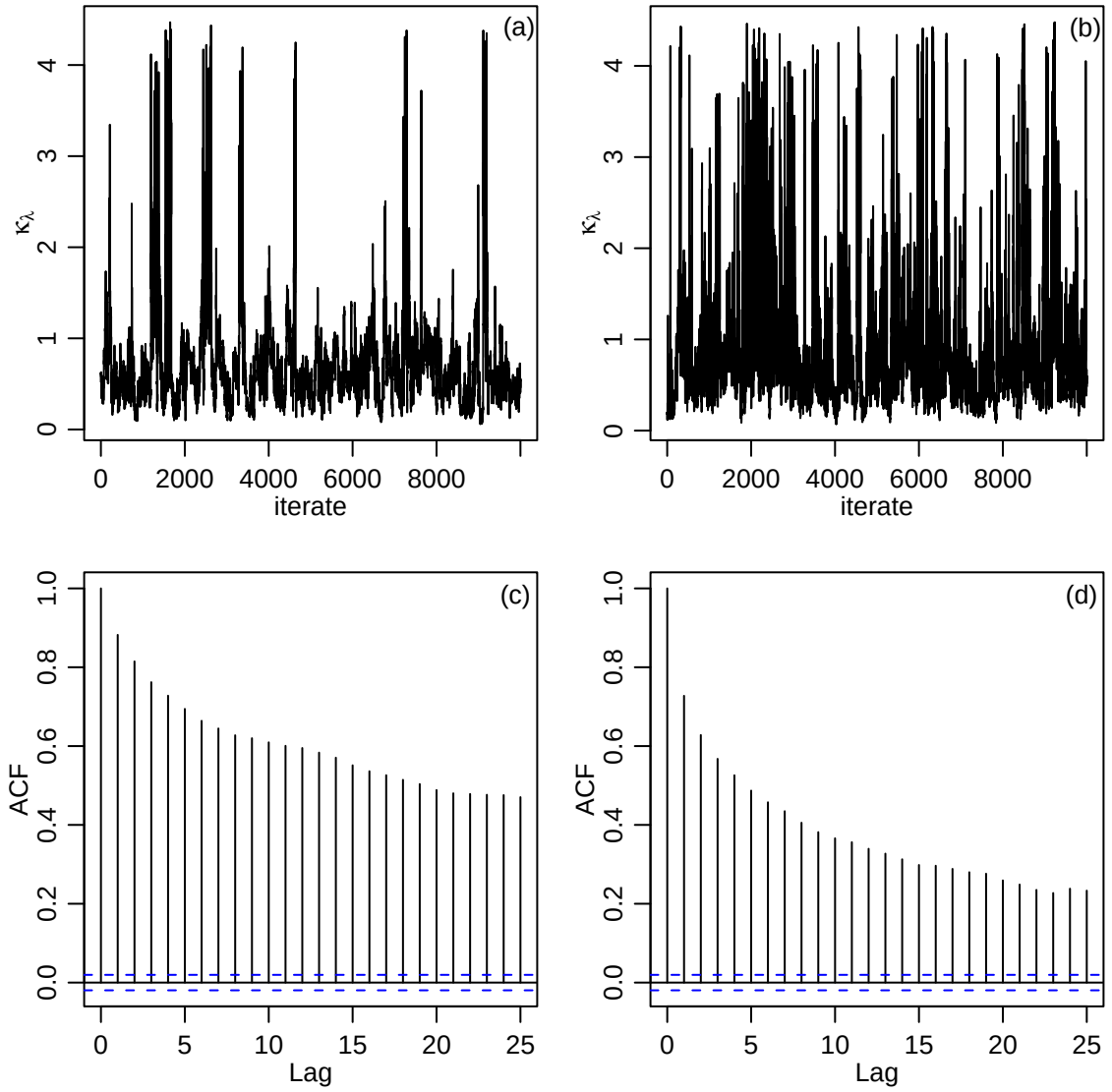

Figure 3.2. Effect of covariate order on κ_λ mixing in a regression problem with $f(x) = \sin\frac{1}{x}$, $x \in [0.1, 0.7]$; the plots are based on subsampling every 10th iteration from chains of length 100,000. I ran two runs, the first with $x_1 = 0.7$, $x_{100} = 0.1$, and proposal standard deviation of 0.24 (acceptance rate of 34%), and the second with $x_1 = 0.1$, $x_{100} = 0.7$, and proposal standard deviation of 0.28 (acceptance rate of 41%). (a) time series plot with $x_1 = 0.7$, (b) time series plot with $x_1 = 0.1$, (c) ACF with $x_1 = 0.7$, (d) ACF with $x_1 = 0.1$.

I use the terms centered and uncentered (Papaspiliopoulos, Roberts, and Sköld (2003) use the term non-centered) to follow the terminology of Gelfand, Sahu, and Carlin (1996), who discuss parameterizations for (generalized) linear mixed models. Their work suggests that when the data are relatively informative about a set of parameters and the priors are uninformative, the model should be reparameterized so that the parameters at the level of the hierarchy directly above the observations are identifiable. In other words, this level should contain parameters that are stochastically (or possibly deterministically) centered about their means rather than containing linear combinations of parameters from which the data will have a hard time distinguishing the parameters. In the alternatives above, we can see that the uncentered parameterization (3.5-3.6) takes $\mathrm{E}\boldsymbol{Y} = \mu + \sigma L(\kappa)\boldsymbol{\omega}$ as a linear combination of parameters, and the data are not able to inform the individual parameters. Only when the prior is informative and the likelihood relatively uninformative is the uncentered parameterization preferred (Gelfand et al. 1996). Papaspiliopoulos et al. (2003) find contrasting situations in which the centered and uncentered are preferred, with a similar conclusion that the centered is better when the likelihood is informative and the uncentered better when the likelihood is relatively uninformative. For a spatial model, they find that the centered outperforms the uncentered in MCMC sampling as the correlation of the spatial process increases.

In my experience, both the centered and uncentered parameterizations lead to slow mixing. In the regression modelling, straightforward sampling of the centered parameterization has the drawback of requiring jitter (Section 3.3), and mixing is even worse than the uncentered parameterization in the example in Section 3.6.2.3. For the uncentered parameterization, one difficulty is that with the generalized Cholesky algorithm, elements of $\boldsymbol{\omega}$ are not used in calculating $\boldsymbol{f}$ when their columns are zeroed out, and the identity of the elements that are unused changes during an MCMC, which may contribute to slow mixing. Note that the discretized parameterization discussed in Section 3.4.2 is similar in structure to the uncentered parameterization and so might be expected to have similar mixing problems. A more important difficulty is that different combinations of $\mu, \sigma, \kappa, \boldsymbol{\omega}$ can give very similar values of $\boldsymbol{f}$. The posterior is probably highly multimodal and moving between these parts of the parameter space is difficult. I concentrate on joint proposals for the function and hyperparameters (as discussed in Section 3.6), because, in my experience, hyperparameter mixing is slow unless the function is adjusted to be consistent with both the proposed

hyperparameter(s) and the likelihood. The proposal scheme becomes even more important in the nonstationary case in which the scalar κ is replaced by the kernels and their attendant spatial processes and hyperparameters. In contrast, in work with spatial random effects in generalized linear mixed models, Christensen and co-workers concentrate on devising proposals for the function (the random effects in their terminology), commenting that the dependence amongst the function values is more of an obstacle than dependence between the function values and the other parameters. Christensen, Møller, and Waagepetersen (2000, 2001) and Christensen and Waagepetersen (2002) report successful mixing of both the function values and the hyperparameters with the uncentered parameterization using Metropolis-adjusted Langevin (MALA) updates for $\boldsymbol{\omega}$, despite using standard Metropolis-Hastings updates for σ and κ. They show that mixing is faster than with the centered parameterization, but their comparison is restricted to a naive implementation of the centered parameterization. This leaves open the question of how Langevin-style updates compare to the PMC proposals (Section 3.6) that I use; in Sections 3.6.2.3 and 4.6.4, I investigate this issue empirically.

The advantage of the uncentered parameterization is that it reparameterizes to a vector of independent variables, $\boldsymbol{\omega}$, and does not require calculation of $L(\kappa)^{-1}$. While the vector $\boldsymbol{\omega}$ has independent components a priori, this is not the case a posteriori, which may explain the difficulties with the uncentered parameterization. As proposed in Christensen, Roberts, and Sköld (2003), to improve mixing, one ideally wants to orthogonalize the parameters with respect to the posterior, adapting to the relative amounts of information in the prior and the likelihood. Papaspiliopoulos et al. (2003) propose a partially non-centered parameterization (PNCP) and Christensen et al. (2003) develop this approach in detail for spatial models, describing a scheme in which a data-dependent reparameterization is done at each MCMC step based on a local transformation that approximately orthogonalizes the parameters with respect to the posterior. They present results that suggest the PNCP works better than either the uncentered or centered alternatives for both the function values and the other parameters. Although this approach seems promising, I do not investigate the PNCP further, in part because implementation requires one to be able to invert $L(\kappa)$ to calculate the data-dependent reparameterization. Modification of the Christensen et al. (2003) approach, in a way that deals with the numerical singularity of my prior covariance matrices, and

application to my model either on its own or in addition to the PMC scheme may help to improve mixing, albeit at some computational cost. My spatial model tends to show slow convergence from poorly chosen starting values, which may be because the chain is not geometrically ergodic (Papaspiliopoulos et al. 2003); the PNCP may do a better job of moving quickly out of the tails of the posterior distribution, which would allow for better assessment of MCMC convergence using multiple chains with disparate starting values.

While the centered and uncentered parameterizations are obviously equivalent probabilistically, there are some interesting distinctions that arise in the context of sampling the parameters via MCMC. In the centered parameterization, one samples the hyperparameters, (μ, σ, κ), and the function $\boldsymbol{f}$. In the uncentered, one samples $\boldsymbol{f}$ only as a function of the other parameters $(\mu, \sigma, \kappa, \boldsymbol{\omega})$, which are the actual parameters in the model. To consider further the relationship between the two parameterizations in the context of MCMC sampling, let's consider the basic model (3.1-3.3) and calculate the likelihood and prior. For simplicity, I assume the error variance, η^2, is fixed.

$$\begin{aligned}
\Pi_{\text{centered}}(\boldsymbol{f}, \mu, \sigma, \kappa | \boldsymbol{y}) &\propto L(\boldsymbol{Y}|\boldsymbol{f}, \mu, \sigma, \kappa)\Pi\left(\boldsymbol{f}|\mu, \sigma, \kappa\right)\Pi(\mu)\Pi(\sigma)\Pi(\kappa) \\
&= \frac{1}{\eta^n}\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{f})^T\left(\eta^2 I\right)^{-1}(\boldsymbol{y}-\boldsymbol{f})\right) \\
&\quad \times \frac{1}{\sigma^n|L(\kappa)|}\exp\left(-\frac{1}{2}(\boldsymbol{f}-\mu)^T\left(\sigma^2 R(\kappa)\right)^{-1}(\boldsymbol{f}-\mu)\right) \\
&\quad \times \Pi(\mu)\Pi(\sigma)\Pi(\kappa) \\
\Pi_{\text{uncentered}}(\boldsymbol{\omega}, \mu, \sigma, \kappa | \boldsymbol{y}) &= L(\boldsymbol{Y}|\boldsymbol{\omega}, \mu, \sigma, \kappa)\Pi(\boldsymbol{\omega})\Pi(\mu)\Pi(\sigma)\Pi(\kappa) \\
&= \frac{1}{\eta^n}\exp\left(-\frac{1}{2}(\boldsymbol{y}-(\mu+\sigma L(\kappa)\boldsymbol{\omega}))^T\left(\eta^2 I\right)^{-1}\right. \\
&\quad \left.\times\,(\boldsymbol{y}-(\mu+\sigma L(\kappa)\boldsymbol{\omega}))\right)\cdot\exp\left(-\frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega}\right)\Pi(\mu)\Pi(\sigma)\Pi(\kappa).
\end{aligned}$$

The key difference between these two parameterizations is the presence in the centered model of the normalizing constant, $(\sigma^n|L(\kappa)|)^{-1}$, of the prior on $\boldsymbol{f}$, which involves the determinant of the prior covariance for $\boldsymbol{f}$. This determinant favors less flexible functions $\boldsymbol{f}$, because when the correlation matrix $R(\kappa)$ has high correlations, the inverse of its determinant is large. This determinant is the way in which Occam's razor (see Section 3.5.1.1) plays a role in the model. Provided a less flexible function is as consistent with the data, based on the likelihood, as a more flexible function, the posterior will favor the less flexible function. The uncentered model does not have this

determinant and seemingly does not favor less flexible functions in the same way that the centered parameterization does. Yet they are the same model; one is a simple reparameterization of the other. The answer to this seeming paradox can be seen in how one might implement a sampling scheme for the centered model. Suppose one were to sample from the centered parameterization, and consider the proposal for κ. If we propose a change to κ while keeping $\boldsymbol{f}$ the same, anything but a small change in κ will be rejected because the current $\boldsymbol{f}$ is correlated in a fashion that is consistent with κ and not the proposal κ^*. This sampling difficulty is precisely why the uncentered parameterization might seem to be a better sampling scheme than the centered with respect to mixing. However, instead of proposing κ alone, let's consider a joint proposal for (κ, $\boldsymbol{f}$). First propose a value κ^*, then propose

$$\boldsymbol{f}^* \sim \mathrm{N}\left(\mu + \sigma L\left(\kappa^*\right)\left(\sigma L(\kappa)\right)^{-1}\left(\boldsymbol{f}-\mu\right), vR\left(\kappa^*\right)\right) \tag{3.7}$$

with v the proposal standard deviation. This seemingly strange construction may make more sense if we consider what happens if $\kappa^* = \kappa$. In that case $\boldsymbol{f}^* \sim \mathrm{N}(f, vR(\kappa))$ namely we are proposing to perturb $\boldsymbol{f}$ in such a way that the new $\boldsymbol{f}^*$ is consistent with the prior correlation matrix. The more complicated joint proposal with κ^* accomplishes the same goal, but with a new value for κ. In (3.7) the inverse of the Cholesky decorrelates the original $\boldsymbol{f}$ and then the decorrelated and demeaned vector $\boldsymbol{\omega} = \left(\sigma L(\kappa)\right)^{-1}\left(\boldsymbol{f}-\mu\right)$ is recorrelated according to $L(\kappa^*)$. How does this bear on the original question of the difference between the centered and uncentered parameterizations? The new joint proposal is a Metropolis-Hastings scheme instead of a Metropolis scheme. Because the proposal is not symmetric, the acceptance ratio now involves the ratio of proposal densities, which I term the Hastings ratio. The Hastings ratio is:

$$\frac{\frac{1}{|L(\kappa)|}\exp\left(-\frac{1}{2}\left(\boldsymbol{f}-\mu-\sigma L(\kappa)\boldsymbol{\omega}^*\right)^T\left(vR(\kappa)\right)^{-1}\left(\boldsymbol{f}-\mu-\sigma L(\kappa)\boldsymbol{\omega}^*\right)\right)}{\frac{1}{|L(\kappa^*)|}\exp\left(-\frac{1}{2}\left(\boldsymbol{f}^*-\mu-\sigma L\left(\kappa^*\right)\boldsymbol{\omega}\right)^T\left(vR\left(\kappa^*\right)\right)^{-1}\left(\boldsymbol{f}^*-\mu-\sigma L\left(\kappa^*\right)\boldsymbol{\omega}\right)\right)} = \frac{\frac{1}{|L(\kappa)|}}{\frac{1}{|L(\kappa^*)|}}.$$

This Hastings ratio exactly cancels the ratio of determinants in the acceptance ratio that comes from the ratio of the posterior terms,

$$\frac{\frac{1}{|L(\kappa^*)|}}{\frac{1}{|L(\kappa)|}},$$

and the result is that we have precisely the acceptance ratio we would have had if we had used the uncentered parameterization, namely an acceptance ratio that does not include the determinant

of the prior covariance matrix. In MCMC, Hastings ratios are required when the proposal makes moves to one part of the space more likely than moves to another part of the space. The preference is corrected in the acceptance ratio by downweighting the portions of the space that are preferred in the proposal. So the smoke clears over the parameterization question, and it is apparent that in the centered parameterization, as previously stated, less flexible functions are favored by the determinant in the prior for the function. In the uncentered parameterization, which can be viewed as the centered parameterization but with a certain type of joint proposal, less flexible functions are favored by biased moves in the proposals of the Markov chain. It is this movement preference that results in a model preference for less flexible functions, provided the data do not argue strongly enough in favor of flexible functions through the likelihood. How does the proposal bias toward less flexible functions come about? Any particular less flexible function is more likely to be sampled than a particular flexible function, because when a κ^* that gives high correlations is proposed, we conditionally sample $\boldsymbol{f}^*$ from a smaller space than the $\boldsymbol{f}^*$ sampled when proposing a κ^* that gives smaller correlations.

Hence it seems clear that sampling from either parameterization is equivalent and furthermore that in evaluating mixing in the uncentered parameterization, one need not be concerned with mixing of the $\boldsymbol{\omega}$ parameters but can safely focus on the values of $\boldsymbol{f}$ and its hyperparameters, since the uncentered parameterization is equivalent to the centered parameterization with joint proposals of $\boldsymbol{f}$ and its hyperparameters. This is reassuring, because my experience suggests that the $\boldsymbol{\omega}$ values mix very slowly. The reason for this may be related to the fact that values of $(\boldsymbol{\omega}, \mu, \sigma, \kappa)$ in disparate parts of the parameter space can give very similar values of $\boldsymbol{f}$. Moving between these disparate regions can be very difficult. While slow mixing of $\boldsymbol{\omega}$ may not be of concern, slow mixing of (μ, σ, κ) is still a concern. In particular, σ and κ are the parameters in the model that control the degree of smoothing.

3.4.2 Discretized parameterization

One important drawback to both parameterizations in the previous section is the need to calculate the Cholesky decomposition of the covariance matrix, which is an O (n^3) operation. In much of his work with the nonstationary kernel convolution covariance, Higdon avoids working with the

covariance matrix and its Cholesky by working with a discretized version of the model (Higdon 1998, 2002). The easiest way to understand this approach is to go back to Higdon et al. (1999)'s construction of the kernel convolution nonstationary covariance, in which the nonstationary correlation function,

$$C(\boldsymbol{x_i}, \boldsymbol{x_j}) = \int_{\Re^2} K_{\boldsymbol{x_i}}(\boldsymbol{u}) K_{\boldsymbol{x_j}}(\boldsymbol{u}) d\boldsymbol{u},$$

is seen as the correlation function of a process constructed as

$$Z(\boldsymbol{x}) = \int K_{\boldsymbol{x}}(\boldsymbol{u}) \psi(\boldsymbol{u}) d\boldsymbol{u},$$

where $\psi(\boldsymbol{u})$ is a Gaussian white noise process. To construct a process with approximately the kernel convolution covariance, consider a discrete grid of independent standard normal random variables as an approximation to the white noise process. Using the spatially-varying kernels, calculate the value of the process $Z(\boldsymbol{x}) = K\boldsymbol{\psi}$ where K is a matrix in which the values in the ith row are the values of the kernel centered at $\boldsymbol{x_i}$ evaluated at the locations of the white noise grid. This constructs the process, $Z(\boldsymbol{x})$, as a weighted average of a discrete set of white noise values, $\boldsymbol{\psi}$. Changing the kernels changes the covariance structure of the process, while different values of $\boldsymbol{\psi}$ give different realizations. This formulation looks identical to basis function regression, with the white noise values being the basis coefficients and the columns of K being the basis functions evaluated at the locations of interest. The advantage of working with this construction of the process is that calculating a realization of the process is O(nm) operations, where m is the number of locations in the white noise grid.

There are several potential disadvantages of the approach. First, one needs to specify the white noise grid. If the grid is too sparse, one will not be able to adequately model the fine-scale covariance structure in the data. If the grid is sufficiently fine, it seems possible that instead of the covariance structure being modelled by the kernels, the covariance structure might be absorbed into $\boldsymbol{\psi}$ during the fitting process. There is no penalty in the multivariate Gaussian prior density for the presence of correlation between elements when independence is specified, only a penalty when correlation is specified and the elements appear to be independent in reality. This could result in incorrect interpretation of the covariance in the data based on the fitted kernels. Another difficulty is one I observed in practice; it can be difficult to achieve reasonable mixing of the white noise

random variables, presumably because many different vectors of white noise values can produce very similar values of the process. This concern might be alleviated by an argument along the lines made for disregarding mixing of the elements of $\boldsymbol{\omega}$ in the previous section, but it is a question that should be addressed. Finally, in high dimensions, the number of grid locations needed to cover the space grows quickly and the computational advantages of the discretization may diminish. For the regression model of Chapter 4, the optimal white noise grid would change with the type of data, so I do not view the approach as being sufficiently general. For the spatial model of Chapter 5 I did not use the discretization because hyperparameter mixing in the discretized model was slow and with replicated observations, the discrete approach requires a separate process (but the same kernels) for each replicate, diminishing the computational advantage.

A general discretized kernel convolution can be done by constructing a process through the convolution of a white noise (or other) process with kernels of arbitrary functional form in place of the Gaussian kernels. Mirroring the construction of the generalized kernel convolution covariance in Chapter 2, one might construct a process with each kernel being a scale mixture of kernels. For example, the Matérn correlation function has the same functional form as the Bessel density (McLeish 1982), which is a scale mixture of Gaussian densities. The difficulty in producing a process with a discretized version of the nonstationary Matérn correlation given in Chapter 2 is that the generalized kernel convolution covariance is based on mixing the product of two Gaussian kernels over the same scale parameter, S,

$$\begin{aligned} C(\boldsymbol{x_i}, \boldsymbol{x_j}) \;\propto\; & \int\int \exp\left(-\frac{1}{2}\left(\boldsymbol{x_i}-\boldsymbol{u}\right)^T\left(\frac{\Sigma_i}{s}\right)^{-1}\left(\boldsymbol{x_i}-\boldsymbol{u}\right)\right) \\ & \times \exp\left(-\left(\boldsymbol{x_j}-\boldsymbol{u}\right)^T\left(\frac{\Sigma_j}{s}\right)^{-1}\left(\boldsymbol{x_j}-\boldsymbol{u}\right)\right) d\boldsymbol{u}dH(s), \end{aligned} \tag{3.8}$$

while the covariance function generated by convolving two Bessel densities involves two independent scale parameters:

$$\begin{aligned} C(\boldsymbol{x_i}, \boldsymbol{x_j}) \;\propto\; & \int\int\int \exp\left(-\frac{1}{2}(\boldsymbol{x_i}-\boldsymbol{u})^T\left(\frac{\Sigma_i}{s_1}\right)^{-1}\left(\boldsymbol{x_i}-\boldsymbol{u}\right)\right) \\ & \times \exp\left(-(\boldsymbol{x_j}-\boldsymbol{u})^T\left(\frac{\Sigma_j}{s_2}\right)^{-1}\left(\boldsymbol{x_j}-\boldsymbol{u}\right)\right) d\boldsymbol{u}dH_1(s_1)dH_2(s_2) \\ = \;& \int B((\boldsymbol{x_i}-\boldsymbol{u})^T(\Sigma_i)^{-1}(\boldsymbol{x_i}-\boldsymbol{u}))\cdot B((\boldsymbol{x_j}-\boldsymbol{u})^T(\Sigma_j)^{-1}(\boldsymbol{x_j}-\boldsymbol{u}))d\boldsymbol{u}, \end{aligned} \tag{3.9}$$

where B is the Bessel density function. Since (3.8) and (3.9) are not equivalent, it is not clear how one would construct a discretized version of the nonstationary Matérn covariance given in Chapter 2.

3.5 Model Dimensionality and Parameter Identifiability

In this section, I discuss how functions of disparate flexibility are embedded in the Gaussian process prior without an explicit change in model dimensionality. The Gaussian process model moves continuously between less flexible and more flexible functions; by comparison free-knot spline models change dimension in discrete jumps as knots are added and deleted. I describe how the GP regression model implicitly favors less flexible functions, provided they are consistent with the data. While the model has constant nominal dimension, I present one way of estimating the implicit dimension for a given set of hyperparameters. This allows me to put an explicit prior over function flexibility. Finally, while embedding functions with different degrees of flexibility in a single structure is an attractive notion, it has implications for interpreting, parameterizing, and sampling the parameters of the model.

3.5.1 Smoothing and dimensionality in the GP model

3.5.1.1 Occam's razor

As I alluded to earlier in the chapter, many Bayesian nonparametric regression models implicitly penalize more complicated models, which produce more flexible functions. Simpler models that are consistent with the data receive higher prior density because the simpler models spread their probability mass over a smaller function space of less flexible functions, and a correspondingly smaller data space, than do complicated models (Rasmussen and Ghahramani 2001; Denison et al. 2002). In the free-knot spline models, models with more knots spread their probability mass over larger spaces than models with fewer knots. The general effect, called Occam's razor (for the notion that simpler models should be preferred, all else being equal), is most easily understood by considering the marginal likelihood of the data, having integrated the appropriate parameters out of the model. To investigate this in the Gaussian process model, let's integrate the function out of

the model, set $\mu = 0$ and $\sigma = 1$ for simplicity, and consider the posterior for κ:

$$\begin{aligned} \boldsymbol{Y} &\sim \mathrm{N}\left(0, R_{\boldsymbol{f}}(\kappa) + C_{\boldsymbol{Y}}\right) \\ \Pi(\kappa|\boldsymbol{Y}) &\propto \Pi(\kappa) L(\boldsymbol{Y}|\kappa) \\ &= \Pi(\kappa) \frac{1}{|R_{\boldsymbol{f}}(\kappa) + C_{\boldsymbol{Y}}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{y}^T \left(R_{\boldsymbol{f}}(\kappa) + C_{\boldsymbol{Y}}\right)^{-1} \boldsymbol{y}\right). \end{aligned}$$

Now consider two scenarios: one with a value of κ that gives high prior correlation in $R_f(\kappa)$ and the second a value of κ giving low prior correlation. If the data truly do exhibit correlation based on their covariate values, then the quadratic form in the exponential of the likelihood will not be much worse under a model with high prior correlation than a model with low prior correlation, but the inverse of the determinant in the normalizing constant will favor the simpler model. So long as the prior, $\Pi(\kappa)$, is relatively uninformative, the posterior for κ will favor values of κ that induce high prior correlation. As we see in Chapter 4, this Occam's razor effect does seem to happen in practice; even when the GP model lacks the prior over implicit model dimensionality discussed in the next section, it does not appear to overfit even though the priors over the hyperparameters do little to differentiate between more and less flexible functions. This result is mirrored by the findings of Biller (2000) and DiMatteo et al. (2002) for free knot spline models in which the prior on the number of knots has little effect on the inference for the regression function, suggesting that the Bayesian framework allows the data to select the appropriate level of smoothing, with limited effect of the user's prior on the amount of smoothing.

3.5.1.2 Model dimension

In linear regression models, it is simple to estimate the dimension of the model. For polynomial regression, there are $k + 1$ parameters, where k is the degree of the highest polynomial, i.e., two for linear regression, three for quadratic regression, etc. The dimension tells us something about the flexibility of the model and the resulting function estimate. For the cubic regression spline model the dimension is taken to be $k + 4$ where k is the number of knots. This is based on the function estimate being $k + 1$ cubic polynomials, with three constraints at each knot ($k + 4 = (k+1) \cdot (3+1) - 3k$). In these models, the dimension changes discretely as the polynomial degree or number of knots change. In the GP model, the flexibility of the function depends on the degree

of correlation in the correlation matrix, R_f, and the value of σ^2. The dimension of the model changes continuously rather than in discrete steps.

A linear smoother is a smoother in which the estimated function can be represented as

$$\tilde{\boldsymbol{f}} = S\boldsymbol{y}.$$

One standard way to estimate the effective number of parameters, or degrees of freedom (df), of a linear smoother is the trace of the smoothing matrix, S (Hastie and Tibshirani 1990, p. 52). This approach is motivated by the special case of polynomial regression, which has smoothing matrix, $S = B\left(B^T B\right) B^T$ where B is the design matrix constructed from the polynomial basis functions. The trace of this matrix is $k+1$, with k the degree of the highest order polynomial in the basis, so for this special case we recover the correct number of parameters in the model. To apply this to the GP regression model, remember that

$$\mathrm{E}(\boldsymbol{f}|\boldsymbol{y}, \theta_f, \eta) = \tilde{\boldsymbol{f}} = \mu + \sigma^2 R_{\boldsymbol{f}}(\sigma^2 R_{\boldsymbol{f}} + C_{\boldsymbol{Y}})^{-1}(\boldsymbol{y} - \mu). \tag{3.10}$$

Therefore, we have

$$df = \mathrm{tr}\left(\sigma^2 R_{\boldsymbol{f}}\left(\sigma^2 R_{\boldsymbol{f}} + C_{\boldsymbol{Y}}\right)^{-1}\right) + 1. \tag{3.11}$$

I add one to account for estimating μ, which is clearly the correct thing to do because in the limit of $\sigma \to 0$ we estimate $\tilde{f} = \mu$, using one parameter for the function and (3.11) gives $df = 1$. For a given set of hyperparameter values, I can estimate the flexibility of the conditional posterior mean function with df. There are alternatives for estimating df (Hastie and Tibshirani 1990), but this approach is widely used and seems adequate for my purposes. It also provides a way to impose one's prior belief about the flexibility of the function on the model. First, place a prior on the parameters of the model. Because of the high dimensionality of the model, it is difficult to know exactly how to choose this prior structure so as to encapsulate one's beliefs about flexibility. Instead of trying to do this painstakingly through the parameters, impose an additional prior constraint that is simply a distribution on df. If the additional prior on df is bounded, the new prior is guaranteed to be proper. While sampling from the Markov chain, for each set of parameter values, you can include in the prior calculation the contribution from the distribution over df. In practice, in one dimension, the model does a good job of choosing the flexibility without a prior over df. However,

in higher dimensions, the model does have a tendency to undersmooth, at least with relatively few observations, so the imposition of the additional df prior helps somewhat (Section 4.6.2).

There are some drawbacks to the approach. It gives information about the function only through the conditional posterior mean, rather than the current sample of the function in the Markov chain. Also, when the smoothness parameter of the Matérn approaches $\frac{1}{2}$ (the exponential correlation function), the sample paths can be very unsmooth locally because of the lack of differentiability, but this is not well-reflected in the value of df; the calculation of df seems to primarily reflect the level of smoothness at coarser scales (what I have been referring to as flexibility).

For non-Gaussian likelihoods, this approach is not directly applicable, but we might apply the df calculation to an approximation to the posterior mean of the same form as (3.10),

$$\mu + \sigma^2 R_{\boldsymbol{f}}(\sigma^2 R_{\boldsymbol{f}} + C'_{\boldsymbol{Y}})(\boldsymbol{y}' - \mu),$$

where $C'_{\boldsymbol{Y}}$ and $\boldsymbol{y}'$ are approximations described in Section 3.6.2.3.

3.5.1.3 Covariate selection

One desirable feature of a regression model is that the model ignore covariates that are unrelated to the response. We want the function to be flat in the direction of the unimportant covariate(s). In the GP model, this corresponds to having the modelled covariance between locations be unaffected by the distance between locations with respect to the unimportant covariate(s). In the kernel-based nonstationary covariance, this is achieved by the kernel size being large in the direction of the unimportant covariate(s), with the appropriate eigenvalues being large. Unfortunately, allowing such large eigenvalues in the model can result in poor mixing during MCMC simulation because the chain tends to wander in that part of the parameter space as the likelihood and prior are relatively flat there. Once the eigenvalues are large, changes to them do not drastically affect the resulting correlations produced by the kernel convolution. Part of the issue here is that I propose vectors of parameters to speed the MCMC and use a single proposal variance; proposing elements of the vectors with different proposal variances would allow for better mixing, particularly if large proposal variances were used when the kernels were large in certain directions. To improve mixing, I limit the size of the eigenvalues to prevent them from wandering off to such extremely large

values. However, if we are in high dimensions and there are one or more unimportant covariates, this limitation on the eigenvalue size means that some influence of the unimportant covariate(s) on the covariance structure remains. This can result in undersmoothing, which I have observed for some simulated datasets in which one or more covariates have no effect on the response variable. One possible way to avoid this problem would be explicitly include covariate selection in the model with a discrete part of the parameter space that represents very large eigenvalues. This effectively makes the function flat in the direction without requiring the eigenvalue to become extremely large. In addition to having to explicitly change model dimension during MCMC, the implementation of this idea may not be straightforward because of the need to have the eigenvalues at different locations be spatially correlated.

3.5.2 Parameter identifiability and interpretability

The continuously changing model dimension that I discuss in Section 3.5.1.2 has advantages and disadvantages. The primary advantage is that the model can move smoothly between structures of different implicit dimension and these structures are part of the same overall model with the same number of explicit parameters, so methods for dealing with models of different dimension, such as RJMCMC or model selection techniques are not required. Two potential disadvantages are that the meanings of the hyperparameters change as the implicit dimension changes and that in certain situations, parameters in the model become unidentifiable, particularly as the implicit dimension decreases. This lack of identifiability occurs in both the centered and uncentered parameterizations, albeit in somewhat different ways. For simplicity I will focus on the simple stationary Gaussian process prior for $f(\cdot)$ with one covariate, but the ideas apply in higher dimensional covariate spaces and to nonstationary priors.

3.5.2.1 Uncentered parameterization

As discussed in Gelfand et al. (1996), there is an inherent lack of identifiability in the uncentered parameterization, which takes

$$\boldsymbol{f} = \mu + \sigma L(\kappa)\boldsymbol{\omega}. \tag{3.12}$$

Different sets of parameter values $(\mu, \sigma, \kappa, \boldsymbol{\omega})$ can give the same value for $\boldsymbol{f}$. This makes it difficult to interpret the parameters and can lead to slow mixing in an MCMC simulation. The problem is particularly acute in the limit as the correlation approaches 1, namely as $\kappa \to \infty$. In this situation, $f(\boldsymbol{x}) = \mu + \sigma\omega_1$, and the data are unable to distinguish between the three parameters that determine the constant function $\boldsymbol{f}$. Also, there is a second way for $\boldsymbol{f}$ to be constant: $\sigma = 0$. In other words, both σ and κ can cause the covariate to have no impact on the response. They approach this limit in different ways because σ causes global smoothing while κ causes local smoothing that becomes global in its effect as κ gets very large.

The lack of identifiability can also make it very difficult to interpret the parameters. Consider σ and $\boldsymbol{\omega}$ in (3.12). We can obtain the same value for $\boldsymbol{f}$ by multiplying σ by a constant and $\boldsymbol{\omega}$ by the inverse of the constant. Since the $\mathrm{N}(0, I)$ prior on $\boldsymbol{\omega}$ strongly rewards small values of the vector-valued $\boldsymbol{\omega}$, σ will be driven up and $\|\boldsymbol{\omega}\|$ down, except to the extent that the prior on σ prevents this. As the number of elements in the vector increases, the magnitude of this effect increases as well. In situations such as this where the only information about a parameter is in the prior, there tends to be parameter drift during MCMC (Gelfand et al. 1996).

These identifiability issues suggest that the uncentered parameterization may be a poor choice, particularly for σ, but there are also issues of identifiability in the centered parameterization.

3.5.2.2 Centered parameterization

To assess identifiability in this parameterization, let's again consider extreme values of σ and κ. As $\kappa \to \infty$, the function becomes constant and we are in the situation of estimating a single value f based on the observations. This is fine for estimating f, but we are also trying to estimate the two hyperparameters, μ and σ, a task for which we have very little information (essentially just the information in $\bar{\boldsymbol{Y}}$). In practice, for large but not extreme values of κ that produce high correlation, we can get samples of σ that are very large. When this happens, there is little restriction on μ, since the values of $\boldsymbol{f}$ can be far from μ, and therefore the samples of μ tend to be extreme. So one effect of the parameterization is that the variability in μ changes with the value of σ, as can be seen clearly in time series plots of the MCMC samples of the two hyperparameters from the Bernoulli data example in Figure 3.4.

Next, let's consider the relationship between σ and κ. As in the uncentered parameterization, both $\sigma \to 0$ and $\kappa \to \infty$ can force the function to be constant. This aspect of the model seems undesirable and in practice, I limit how large κ can become, which helps to improve mixing during MCMC simulations. As I mentioned previously, this restriction on κ (which becomes a restriction on the eigenvalues of the kernels in the nonstationary model) does cause difficulties in higher dimensions with respect to covariate selection and undersmoothing.

The relationship between σ and κ, and more generally between the variance and correlation components of a covariance model is a complicated one. In both stationary and nonstationary settings, I have found nonintuitive results and tight correlations between variance and correlation parameters (also noted in Christensen et al. (2003)) that can slow mixing and confuse interpretation. I have not investigated the issue in the nonstationary model, but presumably the issue arises there in a more complicated fashion involving the kernel matrix eigenvalues. For the stationary GPs that I have used, in both the spatial and stationary regression models, I have found very high correlation between the σ and κ parameters, which leads to very slow mixing in the joint space of the two parameters. The correlation seems to occur because similar function values are reasonable according to the prior on $\boldsymbol{f}$ when both σ and κ are large and also when the two are both small. Large values of σ allow the data to still be consistent with a model of high correlations (large κ). I have addressed this issue to some extent in the sampling of the spatial model by jointly sampling the two parameters with $\log \sigma$ a linear function of $\log \kappa$ plus some added noise, and then also sampling $\log \sigma$ on its own. This corresponds quite closely to reparameterizing the two parameters as their sum and difference and to a reparameterization suggested in Christensen et al. (2003). With an inverse gamma prior on σ^2 it is possible to integrate σ out of the model, which might be expected to improve mixing as found in Huerta, Sansó, and Guenni (2001) and Sansó and Guenni (2002). However, this is not the case for my spatial model, as mixing of κ is still slow. This seems to occur because σ is acting in part as an auxiliary variable that allows the parameters to move more easily about the space. In particular, there seems to be a large part of the space with relatively low prior density that corresponds to large values of κ. This large low-density region is sampled poorly when σ is integrated out of the model. There are also unresolved issues in understanding the decomposition of covariance into variance and correlation in high-dimensional modelling of joint

normality that I discuss in Section 5.7.2. Inadequacies in the correlation model appear to force high variance values. This effect seems similar to the correlation between σ and κ that I describe above.

In the regression modelling, I have dealt in part with these issues by fixing σ in the stationary GP priors for the eigenvalues used to construct the kernels, since I know the approximate range of reasonable eigenvalues. Neither kernels that are small relative to the smallest inter-observation distance nor kernels that are much larger than the largest inter-observation distance are desirable. For the eigenvector processes, I take $\sigma = 1$ without loss of generality, because I only use the processes to determine the directions of the eigenvectors and ignore the magnitude information.

While I have been able to use MCMC to sample from the GP models described in this thesis and achieve reasonably good mixing of the hyperparameters and their processes in some cases, slow mixing is an important limitation, and MCMC methods for Gaussian process models are not developed to the point that they can be automated and used widely without extensive user assessment and intervention.

3.6 MCMC Sampling Schemes for GP Models

In this section I discuss possible sampling schemes for MCMC fitting of GP models. After discussing a variety of approaches, I describe a particular joint proposal of process and hyperparameters, which I call posterior mean centering (PMC), that achieves faster mixing of the hyperparameters of the Gaussian process and avoids inversion of the generalized Cholesky factor of the covariance matrix. I use the centered parameterization (Section 3.4.1), but the PMC scheme gives the model the flavor of the uncentered parameterization. For simplicity, I base the discussion here on the simple model (3.1-3.3).

3.6.1 Integrating the process out of the model

If the likelihood is Gaussian or if the Gaussian process is embedded in a model such that there is conjugacy, one can integrate the process out of the model. In the simple model (3.1-3.3), it is straightforward to see that the marginal likelihood (with respect to the process) is

$$\boldsymbol{Y} \sim \mathrm{N}(\mu, \eta^2 I + \sigma^2 R_{\boldsymbol{f}}(\kappa)).$$

The hyperparameters are now directly involved in the likelihood and can be sampled in this lower dimensional parameter space. This is particularly advantageous if the number of locations is large. To estimate or sample from the process, one uses the conditional distribution of the process given the data and hyperparameters, which for the observed locations is normal with

$$\begin{aligned} \mathrm{E}\left(\boldsymbol{f}|\boldsymbol{y},\eta,\mu,\theta\right) &= C_{\boldsymbol{f}}\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}\boldsymbol{y}+C_{\boldsymbol{Y}}\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}\mu \\ &= \mu+C_{\boldsymbol{f}}\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}\left(\boldsymbol{y}-\mu\right) \\ \mathrm{Cov}\left(\boldsymbol{f}|\boldsymbol{y},\eta,\mu,\theta\right) &= \left(C_{\boldsymbol{f}}^{-1}+C_{\boldsymbol{Y}}^{-1}\right)^{-1} \\ &= C_{\boldsymbol{f}}\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}C_{\boldsymbol{Y}}. \end{aligned} \tag{3.13}$$

In an MCMC simulation, one can conditionally sample the process at any chosen iteration of the chain based on the values of the hyperparameters in that iteration. Unfortunately, if there are many locations, computing the conditional variance (3.13) can be computationally intensive, since it involves applying the matrix inverse to $m+n$ vectors of length $m+n$, where n is the number of observed locations and m the number of locations at which one wishes to predict, whereas computation of the conditional mean, which is all that is required in the PMC proposal of Section 3.6.2.2 only requires applying the inverse to one vector, $\boldsymbol{y}-\mu$. If one also wishes to predict at m additional locations, the amount of computation increases even further. One particular advantage of integrating out the process is that instead of having to invert $C_{\boldsymbol{f}}$, computation is done with $\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}$, which for even relatively small error variance, is numerically non-singular. Gelfand, Kim, Sirmans, and Banerjee (2003) take the approach of integrating out the process from within a hierarchical model for housing prices as a function of geographic location and fit the hyperparameters by slice sampling to better handle parameter correlation. Paulo (2002) also integrates out the process and uses a multivariate t proposal density, with scale based on the observed Fisher information, for the hyperparameters. It is also possible to integrate some of the scalar parameters out of the model, in particular, μ, if it has a normal prior distribution, and σ^2, if it has an inverse gamma prior. In sampling spatiotemporal models, Huerta et al. (2001) and Sansó and Guenni (2002) integrate out both the process and σ^2 in order to sample from the marginal distribution of κ. As I discussed in Section 3.5.2, the posteriors for σ and κ can be highly correlated in certain models so integrating

out σ may allow for faster parameter mixing, although for the spatial model, I have found that a joint proposal for the two parameters does a better job than the integration approach.

Other researchers avoid having to sample the hyperparameters by fixing them at reasonable values. In kriging, κ is usually chosen based on the empirical semivariogram, using a fitting method such as maximum likelihood or a more ad hoc procedure (Cressie 1993). In the machine learning literature, many researchers integrate the function out of the posterior, find the hyperparameter values that maximize the posterior, plug these fixed hyperparameters into the model, and then perform inference on the function (Gibbs and MacKay 1997; Vivarelli and Williams 1999). This approach tends to rely on using derivative information to perform the maximization, a subject I will address in Section 3.6.2.1. Even in fully-defined Bayesian models, many researchers have chosen to fix hyperparameters because of mixing problems or the computational intensity of fitting the full model (Higdon 1998; Swall 1999). This approach of fixing the hyperparameters may be satisfactory in many situations, but will underestimate the uncertainty in the resulting estimates. If the values are chosen poorly, the resulting estimates may be strongly biased relative to those in which the hyperparameters are fully sampled.

3.6.2 Methods for sampling the process values

Integrating the process out of the model is very common and successful and is part of the reason for the focus on Gaussian likelihood GP models, since model fitting eases considerably. However, there are many situations in which a Gaussian likelihood is not reasonable or one wishes to embed a Gaussian process prior in a complicated model out of which the process cannot be integrated. In recent years, since the introduction of generalized linear models (McCullagh and Nelder 1989), much attention has been paid to generalizing Gaussian likelihood methods to other situations, with attention often focusing on count data (Poisson likelihoods) and binary data (binomial likelihoods). One example of this is the use of spline models for non-Gaussian likelihoods (Biller 2000; DiMatteo et al. 2002). Diggle et al. (1998) discuss the same generalization for kriging methods for spatial data. This situation also arises in generalized linear mixed models (Breslow and Clayton 1993) that include a spatial random effect (Christensen and Waagepetersen 2002). In this thesis, non-conjugacy arises with the GP priors for the eigenprocesses that determine the kernels in the

nonstationary GP regression model and in the residual variance process for the spatial model. In the machine learning literature, attention has focused on using Gaussian processes for classification (Neal 1996; MacKay 1997; Williams and Barber 1998).

Sampling these various GP-based models requires sampling process values that cannot be integrated out of the model, and this remains an ongoing research challenge because of parameter correlation (Diggle et al. 1998; Gelfand et al. 2003). Approaches that use gradient information, such as the Langevin algorithm, can help in the mixing of the process values, but hyperparameter mixing can still be slow. More exotic versions of MCMC may help in getting the process and hyperparameters to both mix, but the root of the problem is that different regions of the hyperparameter space can produce reasonable process values yet these areas can be hard to find and move between. The crux of the matter lies in being able to move about in hyperparameter space while sampling process values that are consistent with both the hyperparameters and the data. In Section 3.6.2.2 I outline the approach of posterior mean centering to move in hyperparameter space while automatically proposing process values that are more consistent with both the prior and the likelihood. Another challenge is computational speed. Full sampling of GP models is slow, particularly with many observations or many locations at which one wants to predict, because calculations with the covariance matrix are $\mathrm{O}(n^3)$. In Section 3.7 I review some of the work on computation in GP models; most of the research on this topic is being done in the machine learning community.

3.6.2.1 Derivative-based methods

The usual Metropolis-Hastings algorithm does not make use of information from the posterior in choosing proposals. In particular, the gradient of the posterior at the current parameter values may contain valuable information about the direction in which one should sample to move to regions of the parameter space with higher density. The Metropolis-adjusted Langevin Algorithm (MALA), also known as Langevin-Hastings, uses the gradient of the posterior in making proposals and includes in the Hastings ratio the appropriate correction needed because the Langevin diffusion is discretized (Robert and Casella 1999, Section 6.5.2; Christensen et al. 2001). In the context of GP models, Langevin-style proposals are most helpful in proposing the process values, rather than scalar parameters (Roberts and Rosenthal 1998; Christensen and Waagepetersen 2002).

However, Christensen et al. (2000) show that the hyperparameters mix well when the process is proposed using the Langevin approach and the scalar hyperparameters are proposed using standard Metropolis, albeit still requiring at least 100,000 MCMC iterations for only 70 spatial locations. The Langevin approach tends to speed the movement of the chain toward the modes of the process (Robert and Casella 1999). Christensen et al. (2003) use a partially non-centered proposal (PNCP) with Langevin updates and report greatly improved mixing for both the hyperparameters and the process values. Christensen and co-workers (prior to Christensen et al. (2003)) use the uncentered parameterization (3.6) with parameter $\boldsymbol{\omega}$. In the generalized linear mixed model framework, the Langevin proposal for $\boldsymbol{\omega}$ is

$$\boldsymbol{\omega}^* \sim \mathrm{N}\left(\boldsymbol{\omega} + \frac{v^2}{2}\nabla(\boldsymbol{\omega}), v^2 I\right), \tag{3.14}$$

where

$$\begin{aligned}\nabla(\boldsymbol{\omega}) &= \frac{\partial}{\partial \boldsymbol{\omega}} \log \Pi(\boldsymbol{\omega} \mid \boldsymbol{y}, \mu, \sigma, \kappa, \eta) \\ &= -\boldsymbol{\omega} + \frac{1}{\eta^2}\sigma L_{\boldsymbol{f}}(\kappa)^T(\boldsymbol{y} - \boldsymbol{g})\end{aligned}$$

and $\boldsymbol{g}$ is the mean function (simply $\boldsymbol{f}$ in the normal likelihood case). The Hastings ratio is:

$$\frac{\exp\left(-\frac{1}{2v^2}\left(\boldsymbol{\omega} - \left(\boldsymbol{\omega}^* + \frac{v^2}{2}\nabla\left(\boldsymbol{\omega}^*\right)\right)\right)^T\left(\boldsymbol{\omega} - \left(\boldsymbol{\omega}^* + \frac{v^2}{2}\nabla\left(\boldsymbol{\omega}^*\right)\right)\right)\right)}{\exp\left(-\frac{1}{2v^2}\left(\boldsymbol{\omega}^* - \left(\boldsymbol{\omega} + \frac{v^2}{2}\nabla(\boldsymbol{\omega})\right)\right)^T\left(\boldsymbol{\omega}^* - \left(\boldsymbol{\omega} + \frac{v^2}{2}\nabla(\boldsymbol{\omega})\right)\right)\right)}.$$

To compare mixing based on the Langevin algorithm to that with the PMC approach (Sections 3.6.2.3 and 4.6.4), I modify this algorithm for the regression and spatial models in this work. I use the centered parameterization, so $\boldsymbol{f}$ is the parameter, but I follow Christensen and co-workers in making use of the derivative of $\boldsymbol{\omega}$ rather than the derivative of $\boldsymbol{f}$. I do this for two reasons; first, Møller, Syversveen, and Waagepetersen (1998) report that the Langevin algorithm performs better when based on $\boldsymbol{\omega}$, and second, the derivative of $\boldsymbol{f}$ involves the inverse of the Cholesky of the covariance, which I cannot generally calculate because of the numerical singularity of the covariance matrix. For the regression model (the proposal for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the spatial model would be similar), the Langevin proposal for $\boldsymbol{f}$ based on (3.14) is

$$\boldsymbol{f}^* \sim \mathrm{N}\left(\boldsymbol{f} - \frac{v^2}{2}(\boldsymbol{f} - \mu) + \frac{v^2}{2}\frac{\sigma^2 R_{\boldsymbol{f}}(\boldsymbol{y} - \boldsymbol{g})}{\eta^2}, v^2 R_{\boldsymbol{f}}\right),$$

and the Hastings ratio, expressed partly in terms of $\boldsymbol{\omega}$ for simplicity, is

$$\frac{\exp\left(-\frac{1}{2v^2}\left(\boldsymbol{\omega}-\boldsymbol{\omega}^*+\frac{v^2}{2}\boldsymbol{\omega}^*-\frac{v^2}{2}\frac{\sigma L_{\boldsymbol{f}}^T(\boldsymbol{y}-\boldsymbol{g}^*)}{\eta^2}\right)^T\left(\boldsymbol{\omega}-\boldsymbol{\omega}^*+\frac{v^2}{2}\boldsymbol{\omega}^*-\frac{v^2}{2}\frac{\sigma L_{\boldsymbol{f}}^T(\boldsymbol{y}-\boldsymbol{g}^*)}{\eta^2}\right)\right)}{\exp\left(-\frac{1}{2v^2}\left(\boldsymbol{\omega}^*-\boldsymbol{\omega}+\frac{v^2}{2}\boldsymbol{\omega}-\frac{v^2}{2}\frac{\sigma L_{\boldsymbol{f}}^T(\boldsymbol{y}-\boldsymbol{g})}{\eta^2}\right)^T\left(\boldsymbol{\omega}^*-\boldsymbol{\omega}+\frac{v^2}{2}\boldsymbol{\omega}-\frac{v^2}{2}\frac{\sigma L_{\boldsymbol{f}}^T(\boldsymbol{y}-\boldsymbol{g})}{\eta^2}\right)\right)}.$$

Following Christensen et al. (2001) and Christensen and Waagepetersen (2002) I truncate the gradient. In the Poisson and binomial cases, we have $\eta = 1$ and $\boldsymbol{f}$ is replaced by the mean functions, $\boldsymbol{g} = \exp(\boldsymbol{f})$ (Poisson) or $\boldsymbol{g} = \boldsymbol{m}\frac{\exp(\boldsymbol{f})}{1+\exp(\boldsymbol{f})}$ (binomial), where $\boldsymbol{m}$ is a vector of the number of trials.

The Langevin algorithm is a special case of an MCMC algorithm called Hybrid Monte Carlo (HMC), which is popular among machine learning researchers. The HMC algorithm (Duane, Kennedy, Pendleton, and Roweth 1987) endows Metropolis-Hastings with not only position information (the current parameter values), but also momentum information, which causes the chain to avoid random walks by favoring movement in the same direction on successive steps through the inertia of the parameters. The method was originally devised in statistical physics and is usually described by analogy to a physical system in which a physical particle is moving through a region of variable potential energy (probability). A number of authors have used the HMC algorithm for sampling from GP-based models and claim success in sampling the model parameters (for Gaussian likelihoods, they apply HMC to the hyperparameters only) (Rasmussen 1996; Williams and Rasmussen 1967; Neal 1997; Williams and Barber 1998; Rasmussen and Ghahramani 2002), although they provide little evidence of mixing to which to compare alternate sampling schemes. The Langevin algorithm is HMC using a single leapfrog step of the discretized position and momentum differential equations.

One drawback to derivative-based methods is that one cannot always obtain a closed form for the derivative of the process values. For example, this is the case for the kernel eigenprocesses in the nonstationary GP regression model (Chapter 4) and for the residual variance process in the spatial model (Chapter 5). It would be possible to calculate numerical derivatives for the process values, but this would be very computationally intensive. Because of this limitation of derivative-based methods and because I have not found that these methods particularly improve hyperparameter mixing, I turn to a different approach in the next section. In assessing the performance of the posterior mean centering proposal scheme, I compare it to both Langevin-style proposals

for the process values and to joint proposals for the centered parameterization that do not include likelihood information (Sections 3.6.2.3 and 4.6.4).

3.6.2.2 Posterior mean centering

The mean centering approach builds on the uncentered parameterization (3.6). Recall that in this parameterization, the parameters (μ, σ, κ) deterministically change $\boldsymbol{f}$ and therefore are directly involved in the likelihood. Because of the deterministic relationship between the parameters and $\boldsymbol{f}$, the process is automatically consistent with the parameter values, in particular with κ, which determines the correlation of $\boldsymbol{f}$. This is because $\boldsymbol{f}$ is produced by filtering the independent (a priori) random variables in $\boldsymbol{\omega}$ through the generalized Cholesky factor, $L_{\boldsymbol{f}}$, which is a function of the current value of κ. The main drawback to the approach and the reason for slow mixing is that proposals for $\boldsymbol{f}$ are not necessarily consistent with the likelihood, except insofar as $\boldsymbol{f}^*$ is similar to $\boldsymbol{f}$ and $\boldsymbol{f}$ is consistent with the likelihood. The PMC approach uses information in the likelihood as well as the prior in making proposals. In the development of the posterior mean centering approach that follows, I will focus on joint proposals for κ and $\boldsymbol{f}$, but the methodology and discussion carry over to μ and σ straightforwardly (as well as to ν in a Matérn parameterization and to the parameters involved in the nonstationary correlation structure).

First I outline a joint sampling scheme for the centered parameterization that is equivalent to the uncentered parameterization, but in which $\boldsymbol{\omega}$ is merely an implicit parameter that is carried along in the calculation. This scheme is the basis for the PMC scheme. Consider a joint proposal for $(\kappa, \boldsymbol{f})$. First propose a value κ^*, then propose

$$\boldsymbol{f}^* \sim \mathrm{N}\left(\mu + \sigma L_{\boldsymbol{f}}\left(\kappa^*\right)\left(\sigma L_{\boldsymbol{f}}(\kappa)\right)^{-1}\left(\boldsymbol{f} - \mu\right), v^2 R_{\boldsymbol{f}}\left(\kappa^*\right)\right). \tag{3.15}$$

As described in Section 3.4.1, the Hastings ratio cancels with the ratio of determinants from the priors in the Metropolis acceptance ratio, and we are left with the acceptance ratio we would have had if we had used the uncentered parameterization and jointly proposed $(\kappa, \boldsymbol{\omega})$. Now let's consider what this proposal for $\boldsymbol{f}$ implies about the proposal for the implicit parameter $\boldsymbol{\omega} = (\sigma L_{\boldsymbol{f}}(\kappa))^{-1}(\boldsymbol{f} - \mu)$.

$$\boldsymbol{\omega}^* \quad = \quad (\sigma L_{\boldsymbol{f}}(\kappa^*))^{-1}(\boldsymbol{f}^* - \mu)$$

$$\begin{aligned} &= (\sigma L_{\boldsymbol{f}}(\kappa^*))^{-1}(\mu + \sigma L_{\boldsymbol{f}}(\kappa^*)\boldsymbol{\omega} + vL_{\boldsymbol{f}}(\kappa^*)\boldsymbol{\psi} - \mu) \\ &= \boldsymbol{\omega} + v\boldsymbol{\psi}, \end{aligned} \tag{3.16}$$

where $\boldsymbol{\psi}$ is the vector of standard normal values used in generating $\boldsymbol{f}^*$. We see that we can update the implicit parameter and calculate the prior for $\boldsymbol{f}$ in the acceptance ratio without using the inverse of the Cholesky:

$$\begin{aligned} \Pi\left(\boldsymbol{f}^*|\mu,\sigma,\kappa^*\right) &\propto \exp\left(-\frac{1}{2}\left(\boldsymbol{f}^*-\mu\right)^T\left(\left(\sigma^2 R_{\boldsymbol{f}}\left(\kappa^*\right)\right)^{-1}\left(\boldsymbol{f}^*-\mu\right)\right)\right. \\ &= \exp\left(-\frac{1}{2}\left(\left(\sigma L_{\boldsymbol{f}}\left(\kappa^*\right)\right)^{-1}\left(\boldsymbol{f}^*-\mu\right)\right)^T\left(\left(\sigma L\left(\kappa^*\right)\right)^{-1}\left(\boldsymbol{f}^*-\mu\right)\right)\right)^T \\ &= \exp\left(-\frac{1}{2}\boldsymbol{\omega}^{*T}\boldsymbol{\omega}^*\right). \end{aligned}$$

This sampling scheme is equivalent to a straightforward sampling scheme for the uncentered parameterization and is therefore the Metropolis-Hastings approach to which Christensen et al. (2000) and Christensen and Waagepetersen (2002) compare their Langevin approach. To allow one to move $\boldsymbol{f}$ separately from κ, I suggest having a separate proposal for $\boldsymbol{f}$ and making v in (3.16) small. In fact, using the development of reversible jump MCMC (Green 1995), one can show that setting $v = 0$ is allowable so long as one includes in the Hastings ratio the Jacobian of the deterministic mapping $\boldsymbol{f}^* = \mu + \sigma L_{\boldsymbol{f}}\left(\kappa^*\right)\left(\sigma L_{\boldsymbol{f}}(\kappa)\right)^{-1}\left(\boldsymbol{f}-\mu\right)$; this Jacobian cancels with the ratios of determinants from the priors, and the result is the same as if $v > 0$.

To begin the development of the posterior mean centering approach, let's consider sampling the simple model (3.1-3.3) with Gaussian likelihood. Conditional on a proposal κ^* and on the data $\boldsymbol{y}$, a Gibbs sample for $\boldsymbol{f}$ is

$$\boldsymbol{f}^* \sim \mathrm{N}\left(\mu + C_{\boldsymbol{f}}\left(C_{\boldsymbol{f}} + C_{\boldsymbol{Y}}\right)^{-1}\left(\boldsymbol{y}-\mu\right), C_{\boldsymbol{f}}\left(C_{\boldsymbol{f}} + C_{\boldsymbol{Y}}\right)^{-1} C_{\boldsymbol{Y}}\right). \tag{3.17}$$

This suggests that if I want a joint proposal for $(\kappa, \boldsymbol{f})$ that gives $\boldsymbol{f}^*$ consistent with both κ^* and the data, I should first sample κ^* and conditional on κ^*, sample $\boldsymbol{f}^*$ from the conditional distribution (3.17). This should allow for large movements in κ space. However, even knowing the conditional distribution (3.17), I may not want to draw a proposal for $\boldsymbol{f}$ from it because it requires calculating the conditional variance and its Cholesky, which is much more computationally intensive than calculating the conditional mean in (3.17). Instead, let's use the conditional mean, but rather than

using the conditional variance, draw the sample so that the variance part of the proposal is consistent with the prior covariance $C_{\boldsymbol{f}}(\kappa^*)$, but not necessarily with the likelihood. The suggested proposal for $\boldsymbol{f}$ uses the conditional mean from (3.17) in place of μ in the proposal (3.15):

$$\boldsymbol{f}^* \sim \mathrm{N}\left(\tilde{\boldsymbol{f}}^* + \sigma L_{\boldsymbol{f}}(\kappa^*)\left((\sigma L_{\boldsymbol{f}}(\kappa))^{-1}(\boldsymbol{f}-\tilde{\boldsymbol{f}})\right), v^2 R_{\boldsymbol{f}}(\kappa^*)\right), \tag{3.18}$$

where $\tilde{\boldsymbol{f}}$ is the conditional mean based on κ, and $\tilde{\boldsymbol{f}}^*$ is the conditional mean based on κ^*. The use of $L_{\boldsymbol{f}}(\kappa)^{-1}$ decorrelates the deviation of the current sample from its conditional mean, and $L_{\boldsymbol{f}}(\kappa^*)$ recorrelates the deviation based on the current correlation parameter so that the proposal is consistent with the current prior covariance. Adding $\tilde{\boldsymbol{f}}^*$ then ensures that the proposal is centered on the new conditional mean for $\boldsymbol{f}$. In Figure 3.3 I give an example of how the sample of $\boldsymbol{f}$ changes when one proposes to move from a large value of κ to a much smaller value.

Assuming that κ is proposed using a Metropolis step, the Hastings ratio for this joint proposal is

$$\frac{\frac{1}{|L_{\boldsymbol{f}}(\kappa)|}\exp\left(-\frac{1}{2v^2}\left(\boldsymbol{f}-\tilde{\boldsymbol{f}}-\sigma L_{\boldsymbol{f}}(\kappa)\boldsymbol{\chi}^*\right)^T\left(L_{\boldsymbol{f}}(\kappa)L_{\boldsymbol{f}}(\kappa)^T\right)^{-1}\left(\boldsymbol{f}-\tilde{\boldsymbol{f}}-\sigma L_{\boldsymbol{f}}(\kappa)\boldsymbol{\chi}^*\right)\right)}{\frac{1}{|L_{\boldsymbol{f}}(\kappa^*)|}\exp\left(-\frac{1}{2v^2}\left(\boldsymbol{f}^*-\tilde{\boldsymbol{f}}^*-\sigma L_{\boldsymbol{f}}(\kappa^*)\boldsymbol{\chi}\right)^T\left(L_{\boldsymbol{f}}(\kappa^*)L_{\boldsymbol{f}}(\kappa^*)^T\right)^{-1}\left(\boldsymbol{f}^*-\tilde{\boldsymbol{f}}^*-\sigma L_{\boldsymbol{f}}(\kappa^*)\boldsymbol{\chi}\right)\right)}$$

$$= \frac{\frac{1}{|L_{\boldsymbol{f}}(\kappa)|}}{\frac{1}{|L_{\boldsymbol{f}}(\kappa^*)|}} \tag{3.19}$$

where $\boldsymbol{\chi} = (\sigma L_{\boldsymbol{f}}(\kappa))^{-1}(\boldsymbol{f}-\tilde{\boldsymbol{f}})$ and $\boldsymbol{\chi}^* = (\sigma L_{\boldsymbol{f}}(\kappa^*))^{-1}(\boldsymbol{f}^*-\tilde{\boldsymbol{f}}^*)$. As in the simpler joint proposal (3.15), the Hastings ratio is just the ratio of the determinants of the Cholesky factors and accounts for the biased movement in the $\boldsymbol{f}$ space caused by the changing size of the $\boldsymbol{f}$ space conditional on κ. Once again, it is allowable to set $v = 0$, namely to use a deterministic proposal for $\boldsymbol{f}^*$ conditional on κ^*, provided that we include the Jacobian of the deterministic mapping, which is the same as the Hastings ratio above (3.19).

A nice feature of the PMC approach is that I can avoid ever calculating $L_{\boldsymbol{f}}(\kappa)^{-1}$ or $|L_{\boldsymbol{f}}(\kappa)|$, which are not possible to calculate with the generalized Cholesky that zeroes out columns. Why is this possible? First, the determinant is not needed because the ratio of determinants in the Hastings ratio (3.19) cancels the ratio of determinants from the prior distribution for $\boldsymbol{f}$. Second, calculation of the inverse is not needed even though it appears the expression for $\boldsymbol{f}^*$. To see why, consider

$$\boldsymbol{\chi} = (\sigma L_{\boldsymbol{f}}(\kappa))^{-1}(\boldsymbol{f}-\tilde{\boldsymbol{f}})$$

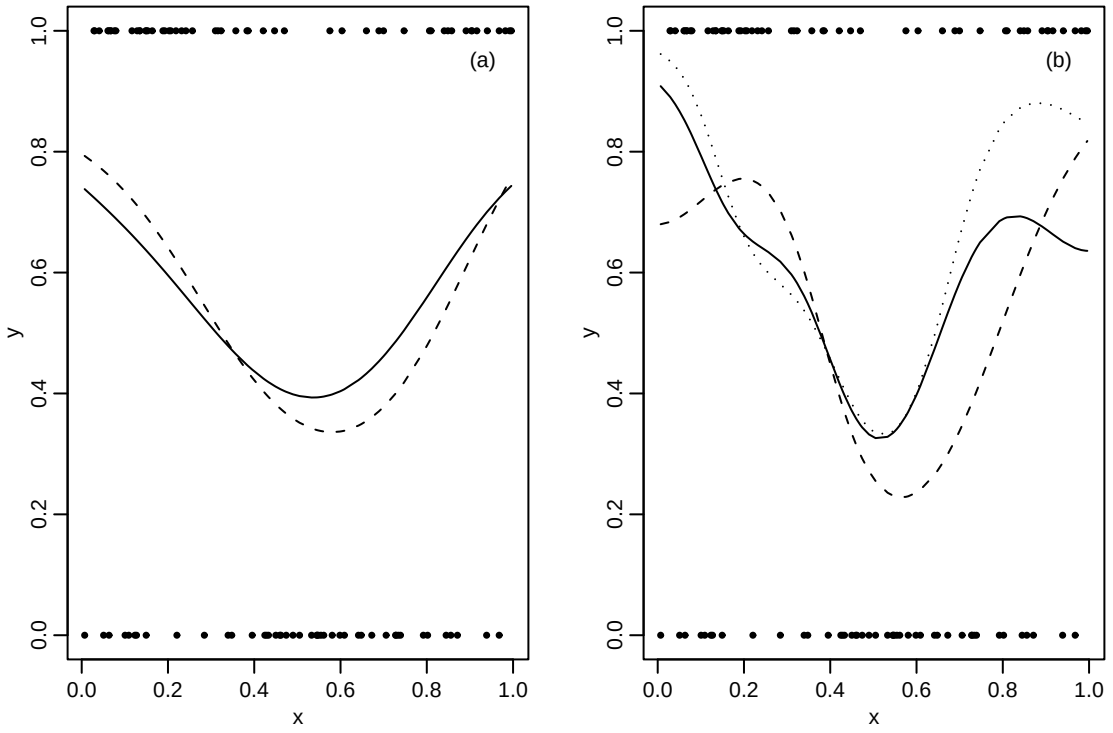

Figure 3.3. Sample function values from an MCMC in a Bernoulli data example with two different values of κ: (a) Sample function (solid line) and conditional posterior mean (dashed line) with $\kappa = 0.70$. (b) Proposing $\kappa^* = 0.30$ and $\boldsymbol{f}^*$ conditional on κ^* using the PMC proposal induces the PMC sample function proposal (solid line) and conditional posterior mean (dashed line). The dotted line is the sample function that would be proposed based on a joint proposal for $(\kappa, \boldsymbol{f})$ without posterior mean centering. Notice that the function proposed without PMC is more extreme than the PMC proposal. Also notice that the conditional posterior mean and sample function proposal are less smooth in (b), but the deviations of the sample function in (a) and the PMC sample function proposal in (b) about their conditional means have similar structure.

$$\begin{aligned} &= (\sigma L_{\boldsymbol{f}}(\kappa))^{-1}\left(\boldsymbol{f}-\mu-C_{\boldsymbol{f}}\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}(\boldsymbol{y}-\mu)\right) \\ &= \boldsymbol{\omega}+\sigma L(\kappa)^{T}\left(C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}\right)^{-1}(\boldsymbol{y}-\mu). \end{aligned}$$

We see first that $\boldsymbol{\chi}$ is a function of the implicit parameter $\boldsymbol{\omega}$, which we carry along and keep current in the MCMC scheme so as to avoid calculating $L_{\boldsymbol{f}}(\kappa)^{-1}$. Second, the only inversion involves not $C_{\boldsymbol{f}}$ alone, which we often cannot do, but $C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}$, for which even relatively small noise variance, η^2, makes $C_{\boldsymbol{f}}+C_{\boldsymbol{Y}}$ numerically non-singular. Using this relationship between $\boldsymbol{\omega}$ and $\boldsymbol{\chi}$ we can move through the MCMC iterations, updating $\boldsymbol{f}$, $\boldsymbol{\omega}$, and $\boldsymbol{\chi}$ without ever needing the inverse of the Cholesky. Prediction at unobserved locations can also be done without inverting $C_{\boldsymbol{f}}$. Let $\boldsymbol{f_1}$ be the function at observed locations and $\boldsymbol{f_2}$ at unobserved locations, with $L'_{\boldsymbol{f}}(\kappa)$ the generalized Cholesky factor of the full correlation of both these sets of locations. We can sample $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ as

$$\begin{pmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \end{pmatrix} = \mu + \sigma L'_{\boldsymbol{f}}(\kappa) \begin{pmatrix} \boldsymbol{\omega} \\ v\boldsymbol{\psi} \end{pmatrix}, \tag{3.20}$$

where $\boldsymbol{\psi}$ is sampled from a multivariate standard normal.

3.6.2.3 Application of posterior mean centering to non-Gaussian data

The posterior mean centering scheme may be of interest in certain cases with Gaussian likelihoods where one does not want to integrate the process out of the model for some reason. But the more important application of the scheme is when the process cannot be integrated out of the model. MCMC mixing in such models can be quite slow (Christensen and Waagepetersen 2002). In that case, we do not know the exact conditional posterior mean, but in certain situations we may be able to approximate it in a way that gives good results. Remember that this is a proposal, and there is no reason we have to use the exact posterior mean, just as in the scheme above we do not use the exact conditional variance for $\boldsymbol{f}$ even though it is available to us. In Chapter 5 I describe such an approximation for the log residual variances in the model. Here I implement the scheme in a generalized regression framework.

For generalized nonparametric regression models in which the error distribution is not assumed Gaussian, the standard GP model takes the following form,

$$Y_i \sim \mathrm{D}\left(g\left(f\left(\boldsymbol{x_i}\right)\right)\right)$$

$$f(\cdot) \quad \sim \quad \text{GP}\left(\mu, \sigma^2 R_{\boldsymbol{f}}(\cdot,\cdot)\right),$$

where D is an appropriate distribution, such as the Poisson for count data or the binomial for binary data, and g is an appropriate inverse link function. In the non-Gaussian case, not only are the observations not Gaussian, but the observations and the process are on different scales because of the link function. In order to propose a new set of values for the process using an approximation to the conditional posterior mean of the form,

$$\mu + C_{\boldsymbol{f}} \left(C_{\boldsymbol{f}} + C_{\boldsymbol{Y}}\right)^{-1} (\boldsymbol{y} - \mu), \tag{3.21}$$

we need all the quantities on the same scale as the process values. Ideally we could apply the link function to the observations, but this is not feasible, as can be seen with the log link for Poisson data when a zero is observed or the logit link for binomial data whenever all successes or all failures are observed. The iteratively-reweighted least squares algorithm (IRLS) (Hastie and Tibshirani 1990) provides a strategy for solving the problem. IRLS expresses each observation as the first two terms in a Taylor expansion of the observation about its mean and uses the variance of the approximation in fitting a weighted least squares regression. I suggest the same approach in the Gaussian process case as a way to devise a PMC algorithm for non-Gaussian data. First express the linearized observation, y_i', through the Taylor expansion,

$$y_i' = g^{-1}(y_i) \approx f(\boldsymbol{x_i}) + \frac{\partial g(\boldsymbol{x_i})}{\partial f(\boldsymbol{x_i})}(y_i - g(\boldsymbol{x_i})).$$

Take the matrix $C_{\boldsymbol{Y}}'$ to be a diagonal matrix, with diagonal elements,

$$\left(C_{\boldsymbol{Y}}'\right)_{ii} = \text{Var}(Y_i') \approx \text{diag}\left(\left(\frac{\partial g(\boldsymbol{x_i})}{\partial f(\boldsymbol{x_i})}\right)^2 \text{Var}(Y_i)\right),$$

and substitute $C_{\boldsymbol{Y}}'$ and $\boldsymbol{y}'$ into (3.21). In the Poisson case, with log link, we have

$$y_i' = f(\boldsymbol{x_i}) + \frac{y_i - g(\boldsymbol{x_i})}{g(\boldsymbol{x_i})},$$

where $g(\boldsymbol{x_i}) = \exp(f(\boldsymbol{x_i}))$, with the diagonal elements of $C_{\boldsymbol{Y}}'$ being $\frac{1}{g(\boldsymbol{x_i})}$. In the binomial setting, with m_i trials, we have

$$y_i' = f(\boldsymbol{x_i}) + \frac{y_i - m_i g(\boldsymbol{x_i})}{m_i g(\boldsymbol{x_i})(1 - g(\boldsymbol{x_i}))}$$

with the diagonal elements of C'_Y being $\frac{1}{m_i g(\boldsymbol{x_i})(1-g(\boldsymbol{x_i}))}$. Using the logit link, we would have $g(\boldsymbol{x_i}) = \frac{\exp(f(\boldsymbol{x_i})}{1+\exp(f(\boldsymbol{x_i}))}$.

The Hastings ratio for this proposal is slightly complicated due to the fact that the value of $\boldsymbol{y'}$ at the current step and the value of $\boldsymbol{y'}$ one would have if one were at the proposed value $\boldsymbol{f}^*$ are different. (We can't use the $\boldsymbol{y'}$ based on $\boldsymbol{f}^*$ because we are still in the process of calculating $\boldsymbol{f}^*$ and that would introduce circularity into the setup.). As a result the Hastings ratio is

$$\frac{\exp\left(-\frac{1}{2}(\boldsymbol{\chi}^{*0} - \boldsymbol{\chi}^{**})^T(\boldsymbol{\chi}^{*0} - \boldsymbol{\chi}^{**})\right)}{\exp\left(-\frac{1}{2}(\boldsymbol{\chi}^{0*} - \boldsymbol{\chi}^{00})^T(\boldsymbol{\chi}^{0*} - \boldsymbol{\chi}^{00})\right)},$$

where

$$\begin{aligned}
\boldsymbol{\chi}^{00} &= \boldsymbol{\omega} + \sigma L_{\boldsymbol{f}}(\kappa)^T(C_{\boldsymbol{f}} + C'_{\boldsymbol{Y}})(\boldsymbol{y'} - \mu) \\
\boldsymbol{\chi}^{0*} &= \boldsymbol{\omega}^* + \sigma^* L_{\boldsymbol{f}}(\kappa^*)^T(C^*_{\boldsymbol{f}} + C'_{\boldsymbol{Y}})^{-1}(\boldsymbol{y'} - \mu^*) \qquad (3.22) \\
\boldsymbol{\chi}^{*0} &= \boldsymbol{\omega} + \sigma L_{\boldsymbol{f}}(\kappa)^T(C_{\boldsymbol{f}} + C'^*_{\boldsymbol{Y}})^{-1}(\boldsymbol{y'^*} - \mu) \qquad (3.23) \\
\boldsymbol{\chi}^{**} &= \boldsymbol{\omega}^* + \sigma^* L_{\boldsymbol{f}}(\kappa^*)^T(C^*_{\boldsymbol{f}} + C'^*_{\boldsymbol{Y}})^{-1}(\boldsymbol{y'^*} - \mu^*). \qquad (3.24)
\end{aligned}$$

The quantities marked by * are calculated based on the proposed values for $\boldsymbol{f}$ and the hyperparameters. Note that in this scheme we once again do not need to use the inverse of the generalized Cholesky. The joint proposal for a new hyperparameter and for $\boldsymbol{f}$ conditional on the hyperparameter involves the following steps. First propose the hyperparameter, either μ^*, σ^*, or κ^*. Next take

$$\boldsymbol{f}^* = \tilde{\boldsymbol{f}}^* + \sigma^* L_{\boldsymbol{f}}(\kappa^*)\left(\boldsymbol{\chi}^{0*}\right),$$

where $\boldsymbol{\chi}^{0*} \sim N(\boldsymbol{\chi}^{00}, v^2 I)$. $\boldsymbol{\omega}^*$ is then calculated based on $\boldsymbol{\chi}^{0*}$ (3.22), and in turn, $\boldsymbol{\chi}^{*0}$ and $\boldsymbol{\chi}^{**}$ are calculated using (3.23) and (3.24). If the proposal is accepted, the new value of $\boldsymbol{\chi}^{00}$ is $\boldsymbol{\chi}^{**}$, not $\boldsymbol{\chi}^{0*}$, since we now know the value $\boldsymbol{f}^*$. Prediction at unobserved locations is done as in the normal likelihood model (3.20).

I compare mixing using the PMC scheme to various other possible sampling schemes on a toy example. I sample $x_i \sim \mathrm{U}(0,1)$ for $i = 1, \ldots, 100$. The response variable is

$$Y_i \sim \text{Bernoulli}\left(p(x_i) = \frac{\exp(f(x_i))}{1+\exp(f(x_i))}\right),$$

where $f(x_i) = \sin(8x_i)$. I take

$$f(\cdot) \sim \text{GP}(\mu, \sigma^2 R(\cdot; \kappa, \nu)),$$

where $R(\cdot; \kappa, \nu)$ is the Matérn correlation function. I compare five sampling schemes:

1. Discrete: I use the parameterization of Higdon (1998) where $\boldsymbol{f} = K\boldsymbol{\omega}$ with $\boldsymbol{\omega} \sim \text{N}(\mu, \sigma^2)$. I take an equally-spaced grid of 30 values for $\boldsymbol{\omega}$ and use the Matérn correlation function as the weight function that determines the elements of K.

2. Uncentered: I take $\boldsymbol{f} = \mu + \sigma L_{\boldsymbol{f}}(\kappa, \nu)\boldsymbol{\omega}$ with $\boldsymbol{\omega} \sim \text{N}(0, I)$ and sample $\boldsymbol{\omega}$ directly rather than $\boldsymbol{f}$.

3. Centered with jitter: I take $\boldsymbol{f} \sim \text{N}(\mu, \sigma^2 R_f(\kappa, \nu))$ and sample in the usual fashion from the hierarchical model, with jitter added to the prior covariance matrix to avoid numerical singularity.

4. Centered with joint sampling: I take $\boldsymbol{f} \sim \text{N}(\mu, \sigma^2 R_f(\kappa, \nu))$ but in sampling any of the hyperparameters, I also sample $\boldsymbol{f}$ conditional on the proposed hyperparameter (as given specifically for κ in (3.15)). In a separate sampling step, I sample $\boldsymbol{f}^* \sim \text{N}\left(\boldsymbol{f}, v^2 R_{\boldsymbol{f}}(\kappa, \nu)\right)$.

5. PMC: I take $\boldsymbol{f} \sim \text{N}(\mu, \sigma^2 R_f(\kappa, \nu))$ but in sampling any of the hyperparameters, I also sample $\boldsymbol{f}$ conditional on the proposed hyperparameter, as given specifically for κ in (3.18) with the modifications to the non-normal likelihood case given at the beginning of this subsection. In a separate sampling step, I sample $\boldsymbol{f}^* \sim \text{N}(\boldsymbol{f}, v^2 R_{\boldsymbol{f}}(\kappa, \nu))$.

For the uncentered, centered-joint and PMC schemes I also ran the MCMC using Langevin sampling applied to the proposals for $\boldsymbol{f}$ ($\boldsymbol{\omega}$ in the case of the uncentered scheme) in the step in which $\boldsymbol{f}$ is sampled separately from the hyperparameters. I had trouble getting the discrete and centered-jitter schemes to perform reasonably when including the Langevin sampling step, so I do not report those results here. The Langevin sampling follows the description given in Section 3.6.2.1, modified as necessary for the sampling schemes above. In adjusting the proposal variances during burn-in, I attempted to achieve the acceptance rates recommended in Roberts and Rosenthal

(2001), namely, 0.44 for scalar parameters, 0.23 for vector parameters, and 0.57 for Langevin updates. I generally came quite close to these rates. Priors are taken to be relatively noninformative, but proper.

I ran the chains for 26000 iterations and retained the last 20000 for assessment of mixing. For many of the schemes, this number of iterations is not even close to sufficient to explore the posterior for some parameters, which I will discuss further in presenting the results. In Table 3.1, I report the relative computational cost for the schemes; the table shows the number of iterations that can be completed for a given scheme relative to completing one iteration of the uncentered scheme, as implemented in the statistical software R.

Table 3.1. Number of iterations that can be completed in the same time as a single iteration of the uncentered scheme.

scheme	number of iterations
discrete	6.79
uncentered	1.00
centered-jittered	0.47
centered-joint	1.01
PMC	0.68

I assess mixing of $\theta \in \{\mu, \sigma, \kappa, \nu, f(0.1), f(0.3), f(0.6), f(0.9)\}$. To evaluate the mixing, I consider time series and autocorrelation function plots, as well as the effective sample size (ESS) approach (Neal 1993, p. 105). In this approach one tries to estimate the effective number of iterations of the chain after accounting for the autocorrelation of the samples. The ESS is defined as

$$ESS \equiv \frac{K}{1 + 2\sum_{k=1}^{\infty} \rho_k(\theta)}, \tag{3.25}$$

where $\rho_k(\theta)$ is the autocorrelation at lag k for θ. In practice, I truncate the summation at the lesser of $k = 1000$ or the first k such that $\rho_k(\theta) < 0.1$, which means that some of the values for ESS are optimistic. I also considered evaluating the sample precision of the estimates, following the approach in Brockwell and Kadane (2002). However, many of the sampling schemes had not fully

explored the posterior after the 20000 iterations, leading to misleading estimates of the precision relative to schemes that more fully explored the posterior. In Table 3.2, I present the ESS values (the maximum possible is $K = 20000$) and in Figure 3.4 I show time series plots for μ, σ, and κ for the five schemes without the Langevin algorithm.

Table 3.2. Effective sample size (ESS) by sampling scheme for key model parameters. $\bar{f}$ is the mean ESS for the function values, averaging over $f(x)$ for all 100 values of x.

scheme	μ	$\log\sigma$	$\log\kappa$	ν	$f(0.1)$	$f(0.3)$	$f(0.6)$	$f(0.9)$	$\bar{f}$
discrete	1984	13	34	1378	496	348	149	456	415
uncentered	29	73	220	659	303	418	116	239	239
uncentered (Lang.)	33	116	111	971	1134	506	426	446	589
centered-jitter	125	12	13	20	129	87	22	152	94
centered-joint	33	47	174	980	197	434	102	220	233
cen.-joint (Lang.)	41	125	201	993	670	493	382	538	477
PMC	2225	288	646	1132	466	390	254	378	356
PMC (Lang.)	1715	422	874	1097	771	846	501	737	674

These indicate that when considering the parameters as a whole the PMC scheme is clearly the best. In particular, for $\log\sigma$ and $\log\kappa$ PMC handily outperforms all of the other methods. In comparing mixing of the function values, without the Langevin approach, many of the methods are relatively similar. The Langevin approach improves the mixing of $\boldsymbol{f}$ for the uncentered, centered-joint, and PMC schemes and seems to somewhat improve mixing for some of the hyperparameters. Table 3.2 does not account for the differing computational efficiencies of the methods. If we adjust by replacing K in (3.25) with the number of samples one could draw for each scheme in the time that it would take to draw 20000 for the PMC scheme, we get the adjusted ESS values in Table 3.3. We see that the results have changed qualitatively in two regards. First, the centered-joint and uncentered parameterizations are competitive with PMC with respect to the function values, but still not with respect to the hyperparameters. Second, because the discrete scheme is much faster than any other scheme, for many of the parameters, the discrete scheme now seems to mix

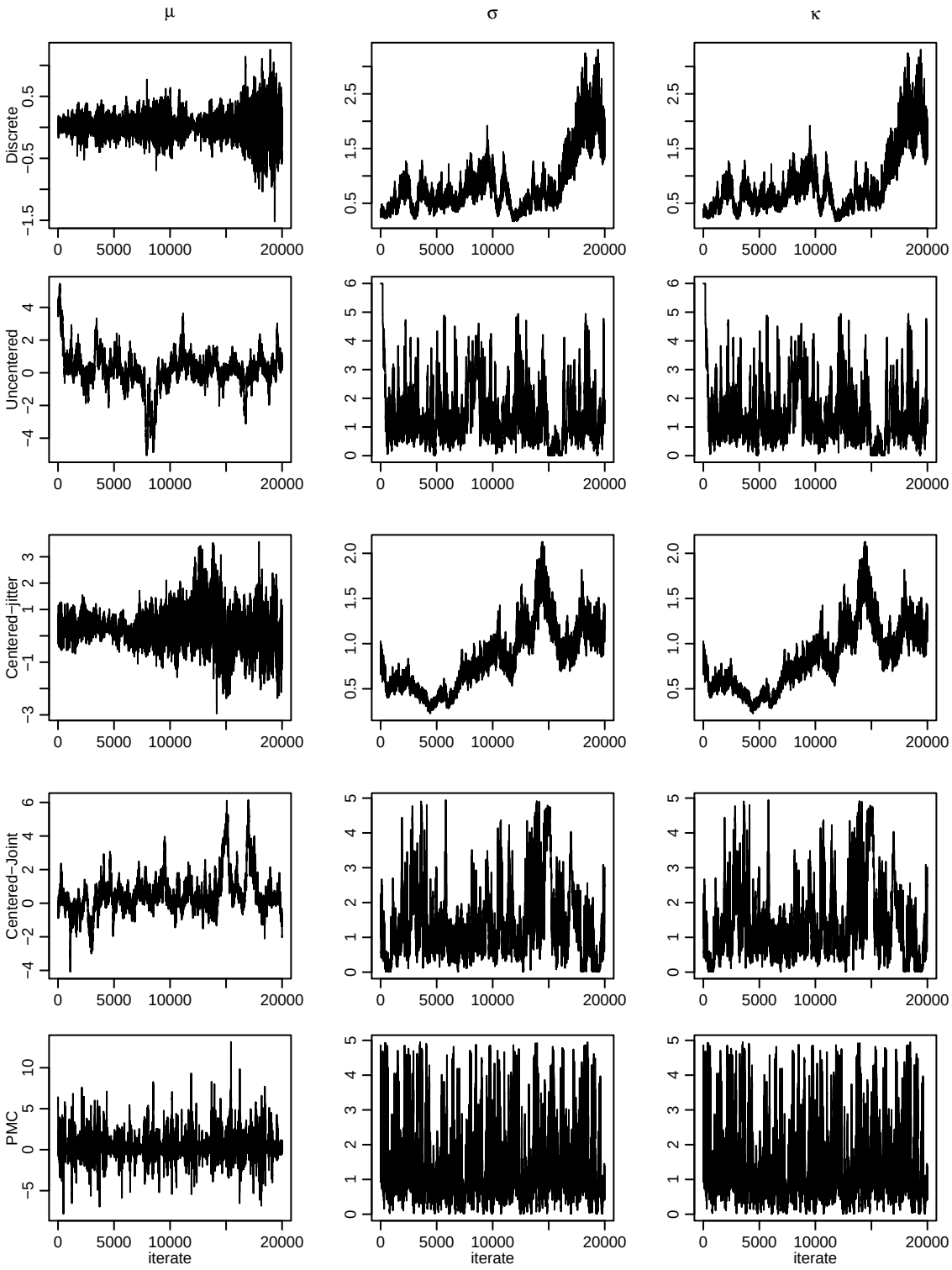

Figure 3.4. Time series plots of μ, σ, and κ for the five basic sampling schemes..

Table 3.3. ESS by sampling scheme for key parameters, adjusted for computational speed. $\bar{f}$ is the mean ESS for the function values, averaging over $f(x)$ at all 100 values of x.

scheme	μ	$\log\sigma$	$\log\kappa$	ν	$f(0.1)$	$f(0.3)$	$f(0.6)$	$f(0.9)$	$\bar{f}$
discrete	19924	133	343	13836	4979	3490	1498	4583	4144
uncentered	42	108	325	974	448	617	171	353	351
uncen. (Lang.)	48	171	163	1428	1667	744	627	655	866
centered-jitter	87	8	9	14	90	61	16	106	65
centered-joint	48	70	258	1458	293	645	152	328	346
cen.-joint (Lang.)	61	186	299	1475	996	732	567	799	708
PMC	2225	288	646	1132	466	390	254	378	372
PMC (Lang.)	1715	422	874	1097	771	845	501	737	674

best. However, this is only the case if we are comfortable with the slow mixing of $\log\sigma$ and $\log\kappa$. In fact, for the discrete scheme the adjusted ESS values for these parameters exaggerate the competitiveness of the scheme. Even the low ESS values in Table 3.3 are optimistic because I cut off the infinite sum at lag 1000. Cutting off the sum at lag 5000 would change the ESS values for the discrete scheme for $\log\sigma$ to 74 and for $\log\kappa$ to 218, which suggests the extremely slow mixing of these parameters is difficult to overcome even with extremely long chains. These parameters are crucial in that they control the flexibility of the regression function, and we seem to be exploring only a portion of the posterior for these parameters in the discrete sampling. Because of this, I would be very uncomfortable using the discrete approach in place of PMC. In sum, the evidence suggests that the PMC sampling scheme dramatically improves the mixing of the hyperparameters relative to the alternative approaches, and that use of the Langevin algorithm in addition to PMC improves the mixing of the function values.

In Section 4.6.4, I demonstrate the success of posterior mean centering with an approximate posterior mean for a binomial data example used in Biller (2000) with similar mixing results to those shown here. In Chapter 5, I use a different PMC scheme in the sampling of the residual

variance field and see much faster, albeit still slow, mixing of the hyperparameters for the variance process. In that case, I use the same approximate posterior mean for the centering at every iteration, because a reasonable approximate mean for each iteration is complicated and makes the Hastings ratio difficult to calculate.

3.7 Computational Challenges of GP Models

Fitting GP models presents computational challenges because of the O(n^3) computations involved in calculations involving the covariance matrices. If one is not fitting the model via MCMC, then analysis of relatively large n, say $n \in (1000, 10000)$, or possibly even larger, is feasible, but when the GP is fit within an MCMC, one needs to perform the matrix calculations repeatedly, so feasible sample sizes fall into the hundreds rather than the thousands. This can be partially alleviated if the process can be integrated out of the model, in which case one can just sample the process conditional on the iterations of the MCMC for the remaining parameters, hopefully a relatively small number of times by subsampling the process at long enough intervals so that the hyperparameter draws are approximately independent. However, even in this situation, the likelihood still involves the marginal covariance matrix of the n observations. Unfortunately, there is no simple closed form for a sample from the process conditional on the observations and the current hyperparameter values. This contrasts unfavorably with the free-knot spline model in which sampling the process does not involve matrix inversion and generally involves a basis function matrix with many fewer basis functions than observations.

In practice, fitting spline-based methods using MCMC is much quicker than fitting the nonstationary GP model via MCMC. However, mixing and convergence are an issue for such competing methods, just as they are for the GP method. (See a discussion of this in Biller (2000) with respect to spline models embedded in generalized linear models.) In particular, the movement between model spaces of differing dimension can make convergence assessment difficult.

3.7.1 Local methods

The primary problem with the GP calculations is that they are global in the sense that the prediction for each observation involves calculation with all the other observations, even those far from the focal observation that have practically no influence upon the prediction. Even if many correlations were exactly zero, prediction at the focal observation still involves the n by n matrix inversion problem. Being able to turn the GP model into one involving local calculations would offer a great computational benefit.

One approach is to divide the covariate space and use local models. Several attempts have been made to use local GPs with their own stationary covariance, and then knit the GPs together, which achieves nonstationarity by using different stationary covariances in different regions. Holmes, Mallick, and Kim (2002) use a Voronoi tessellation of the space and fit stationary GPs in the regions. Rasmussen and Ghahramani (2002) assign individual locations to one of a set of stationary GPs, with the assignment governed by a Dirichlet process and not restricted by location. The key problem here is to choose how to divide the space, which entails its own challenges.

3.7.2 Sparse methods

The local models just described are sparse in the sense that prediction at a focal observation is based only a subset of the observations. Sparsity can also be achieved in other ways. A promising approach currently under investigation in the machine learning community is the use of reduced rank approximations to the covariance matrix, working under the notion that not all of the information in the matrix is required to capture the essence of the dependence structure. There are various methods whose details differ (Smola and Bartlett 2001; Williams and Seeger 2001; Williams, Rasmussen, Schwaighofer, and Tresp 2002; Seeger and Williams 2003). The general approach is to choose a submatrix, C_{mm}, of size m by m from the prior covariance matrix and perform the matrix inversion on the submatrix. Following the development in Seeger and Williams (2003), the reduced rank approximation to the covariance is $C \approx \tilde{C}_{nn} = C_{nm}C_{mm}^{-1}C_{nm}^{T}$ where the subscripts indicate the size of the matrices, based on n observed data points and $m < n$. This corresponds to selecting a subset of the covariates, and the methods differ in how they approach this optimization problem, with tradeoffs between computational speed and optimality. From the basis function per-

spective, using the reduced rank covariance corresponds to using fewer basis functions to represent the function, and the approach can be thought of as a sparse representation in the weight-space. The approach inherently deals with the numerical singularities that arise in GP calculations by explicitly working with the reduced rank covariance.

Some of the reduced rank approaches deal primarily with estimating the posterior mean function conditional on fixed hyperparameters (e.g., Smola and Bartlett (2001)), while Seeger and Williams (2003) suggest optimizing the hyperparameters based on the reduced rank approximation to the marginal likelihood (with respect to the function values), including an approximation of the determinant involved. This is essentially an empirical Bayes approach, but without a prior over the hyperparameters. The approach appears to be generally successful, greatly reducing the computational cost at a limited cost in terms of error, provided the eigenvalues of the covariance decay sufficiently rapidly relative to the error variance and the rank is not reduced too drastically (Williams and Seeger 2001; Williams et al. 2002). At this point, it's not clear how useful the approach would be for MCMC sampling from the full Bayesian model since any approximation to the posterior or marginal posterior will change the stationary distribution, although perhaps not substantively. For the highly-parameterized nonstationary covariance that I employ, it's even less clear how one would proceed, because of the large number of parameters and the need to sample the processes that construct the kernel matrices that determine the nonstationary covariance.

Sparsity in the function-space view involves the use of sparse covariance matrices, namely matrices with many zeroes. Calculations such as the Cholesky decomposition and solutions to linear equations can be done more quickly with sparse matrices than with non-sparse ones, and there are many algorithms and established computer code for performing these calculations. For the covariance matrices in GP priors, if the correlation falls off quickly enough, many elements may be nearly zero. Unfortunately, we cannot just set all the elements below some threshold to zero and still ensure positive definiteness, loss of which would entail being unable to calculate the Cholesky factor or possibly even a reasonable generalized Cholesky factor. An alternative is to enforce sparsity in some other way.

One way is to use kernels that are zero beyond a certain distance (compactly-supported kernels) and calculate the covariance as the convolution of kernels $C(\boldsymbol{x_i}, \boldsymbol{x_j}) = \int_{\Re^P} K_{\boldsymbol{x_i}}(\boldsymbol{u}) K_{\boldsymbol{x_j}}(\boldsymbol{u}) d\boldsymbol{u}$.

The covariance will be positive definite by construction if the kernels are solely a function of their location. The drawback to this approach is that there is likely not a closed form for the covariance, so the integration would have to be done numerically, which would increase the computations in calculating the covariance matrix, possibly more than is gained in using sparse matrix calculations.

An alternative is to use compactly-supported correlation functions (Gaspari and Cohn 1999; Gneiting 1999); one way to create such functions is to multiply a base correlation function by another correlation function that is zero beyond a fixed distance (Gneiting 2001). Nonstationarity is obtained if the base correlation function is nonstationary. However, lack of sample path smoothness based on either correlation function will carry over to the product correlation function since mean square differentiability depends on being able to differentiate the correlation function, hence both terms in the product. The simple cases of correlation functions identically zero beyond a fixed distance, such as the spherical and cubic correlation functions (Abrahamsen 1997), do not give sample paths that are mean square or sample path differentiable. Gneiting (2001) gives the compactly-supported correlation function:

$$R(\tau) = (1-\tau)\frac{\sin(2\pi\tau)}{2\pi\tau} + \frac{1}{\pi}\frac{1-\cos(2\pi\tau)}{2\pi\tau}\,, 0 \leq \tau \leq 1$$

which minimizes $R''(0)$ amongst the functions positive definite on $\Re^3$. If this function is positive definite in higher dimensions, it may also be useful in such cases. I investigated this approach in the spatial modelling but found that the compactifying correlation function not only affects the modelled correlation at long distance scales, but also at short distance scales, where I would like the original base correlation function to be the primary influence. Furthermore, there are limits to the computational gain that can be achieved with sparse covariance matrices, since one is still performing matrix computations with large, albeit sparse, matrices.

3.7.3 Approximate matrix calculations

Much research has focused on efficiently computing with large matrices, in particular approximately solving large systems of linear equations. One method for doing this is the conjugate gradient algorithm (Golub and van Loan 1996, Section 10.2). Such methods may be useful for GP models, although the use of an approximation in the calculation of the prior will in principle

change the stationary distribution of the chain and in practice may therefore give results too far from the real distribution. An important issue in employing such methods is to decide how accurate the solution needs to be and whether this requires so many iterations of the iterative methods used for approximating the solution that the computational savings are meager relative to finding the exact solution. Also, in addition to doing calculations of the form $C^{-1}\boldsymbol{b}$, I need to calculate the determinant of C. Skilling (1989, 1993) discusses methods for doing this, but they seem to be less well-developed than the $C^{-1}\boldsymbol{b}$ approximation. Williams and Barber (1998) tangentially note poor performance with these approximate methods for classification problems. Gibbs and MacKay (1997) find the values of the hyperparameters that maximize the posterior after integrating the process out of the model. Based on the work of Skilling (1989, 1993), they use the conjugate gradient method and an approximation to the trace of the covariance (which is part of the derivative of the determinant) to do the maximization efficiently.

In the fully Bayesian context, treating the hyperparameters as uncertain and approximating both $C^{-1}\boldsymbol{b}$ and the determinant of C leads one to the centered parameterization and the necessity of employing jitter. As I have demonstrated in Section 3.6.2.3, this sampling scheme does not mix well, so computational savings gained in the approximations may be lost in having to run the sampler for more iterations. The other parameterizations and sampling schemes I have outlined make explicit use of the Cholesky of the covariance, but I do not know of any fast approximations for creating the Cholesky factor or calculating $L\boldsymbol{b}$ based only on knowledge of C.

3.7.4 Parallel processing

For problems that justify the extra programming effort, parallel processing offers another alternative to speed the calculations. The most straightforward way to make use of parallel processing is to run multiple chains and combine the results at the end. The one drawback to this is that burn-in must be ensured at the start of each chain, so if the burn-in time is long, much of the parallel processing time can be spent on burn-in iterations rather than on the useable iterations. Also, this requires effort by the user to ensure that burn-in has occurred or to automate the determination. A promising alternative is the technique of Brockwell and Kadane (2002), which splits one chain amongst multiple processors in a clever way. The main issue for employing this technique

is adapting it so it works in the high-dimensional spaces involved when the process itself or the parameters of the nonstationary covariance must be sampled. Finally, there are parallel versions of the Cholesky decomposition (Golub and van Loan 1996, Section 6.6), the construction of which is the primary computational cost of the model. I do not know how large the matrix has to be to make the parallel version more efficient than a non-parallel version, since the communication cost of parallelizing must be amortized.

3.7.5 Non-Bayesian approaches

For large problems in which the computations become an serious impediment, it may be useful to think of non-Bayesian or approximately Bayesian ways of fitting GP models. Maximizing the hyperparameters based on the marginal likelihood and using the conditional distribution of the process is one such approach. It may also be possible to fit the model using classical techniques such as restricted maximum likelihood, described in Higdon (2002), but doing this in the high-dimensional nonstationary model is likely to be difficult.

3.7.6 Fast Fourier transform

Christensen et al. (2000) discuss the use of the FFT to efficiently calculate a matrix square root by embedding the locations under analysis in a larger rectangular grid. This approach may be more applicable for stationary GPs than nonstationary ones, since the nonstationary parameterization I use requires eigenprocesses that determine the kernels and would therefore involve the computational cost of sampling these eigenprocesses and calculating the kernel convolution covariance on the larger grid.

3.7.7 Overview

At this point, I know of no well-established method for efficiently fitting the GP regression model, particularly in a fully Bayesian framework. However, much work is ongoing in this area in the machine learning community. Low-rank approximations to the covariance may make the GP model feasible for large datasets. Whether these approximations or the other approaches described in this

section will be feasible for some representation of the nonstationary covariance model that I have developed is an open question.