Copyright © 1999 by Jenise Lynn Swall
All rights reserved
NON-STATIONARY SPATIAL MODELING USING A PROCESS CONVOLUTION APPROACH

by

Jenise Lynn Swall

Institute of Statistics and Decision Sciences
Duke University

Date: ______________________________

Approved:

________________________
Dr. David Higdon, Supervisor

________________________
Dr. Peter Mueller

________________________
Dr. Robert Wolpert

________________________
Dr. Dean Urban

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Institute of Statistics and Decision Sciences in the Graduate School of Duke University

1999
ABSTRACT

(SUBJECT: Bayesian Statistics)

NON-STATIONARY SPATIAL MODELING USING A PROCESS CONVOLUTION APPROACH

by

Jenise Lynn Swall

Institute of Statistics and Decision Sciences
Duke University

Date: __________________________

Approved:

__________________
Dr. David Higdon, Supervisor

__________________
Dr. Peter Mueller

__________________
Dr. Robert Wolpert

__________________
Dr. Dean Urban

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Institute of Statistics and Decision Sciences in the Graduate School of Duke University

1999
Abstract

This dissertation develops a process convolution approach to modeling the covariance structure of non-stationary Gaussian processes. This methodology is useful in the analysis of many real-life processes in which the usual assumptions of stationarity and/or isotropy are not appropriate. Implementation and related computational issues are also discussed in detail.

Various traditional modeling strategies are available for modeling processes that are stationary, an assumption that is often justifiable and leads to a reasonable analysis. However, some examples seem to require alternative models that can account for a more heterogeneous spatial structure. With a review of some of these procedures, we note that our method has the advantages of providing insight into the extent and nature of the non-stationarity that exists, and presenting this information graphically and clearly. We develop an hierarchical specification for covariance structure that can be used with a single realization from the spatial process and that can provide estimates about uncertainty in the spatial process.

Our approach defines a spatial process $z(s)$ as a convolution of a Gaussian white noise process and a series of convolution kernels $k(s)$. A one-to-one mapping exists between each one of these kernels and its one standard deviation ellipse, even as the kernel is stretched and rotated. We choose the bivariate normal distribution as the form of each kernel, but other forms could be selected instead. These convolution kernels are then specified at each location by a pair of parameters denoting the location of one of the focus points of the ellipse and by another parameter that serves to shrink or expand the kernel.

We use MCMC to explore the posterior distribution of the model’s parameters. This provides a simple way to estimate the variability of the parameters. In con-
junction with this, we provide explanation of the many computational issues that are encountered when employing such a hierarchical covariance structure. This includes some complicated parameter updating mechanisms to help speed the convergence of the parameters to their posterior distributions using MCMC, as well as the development of modified Cholesky algorithm that handles semi-positive definite matrices in a sensible way.

Finally, we summarize the results, comparing the results using the process convolution approach with other, more traditional methods. Possible extensions and directions for future research serve to conclude the discussion.
Acknowledgements

I would like to first thank my advisor, Prof. David Higdon. His support, both professional and personal, has been invaluable during the past four years. His enthusiasm is contagious, and his door is always open for a brief pep talk or a difficult technical discussion.

Many thanks go to Andrew Gallatin. In addition to his generous emotional support, his expertise with computers (in particular, his knowledge of the C programming language and associated debuggers) has proven a tremendous resource.

Thanks also to Prof. Alvin Lebeck for the use of his research group’s computing resources for the completion of the work that formed the basis of this dissertation.

Special thanks are reserved for my family. Thank you for supporting me in all the endeavors I have chosen. When the going got tough, you were always there to hear my complaints; when things were on the upswing, you were always the first to celebrate with me.
Contents

Abstract iv
Acknowledgements vi
List of Figures x
1 Preliminaries 1
2 Introduction of the model 17
  2.1 A general model ........................................... 17
  2.2 Priors .................................................. 19
3 The process convolution approach 25
  3.1 Motivation .............................................. 25
  3.2 Process convolution approach .............................. 26
    3.2.1 The approach in the case of stationarity .......... 26
    3.2.2 The process convolution approach in the case of non-stationarity 29
  3.3 Connection of process convolution idea to the visualization of an ellipse 30
  3.4 Priors for process convolution specific parameters .......... 32
    3.4.1 Priors for foci coordinates \( \psi_x \) and \( \psi_y \) .................... 33
    3.4.2 Priors for coordinate axis shrinking/expansion \( \tau \) ............ 35
4 Posterior exploration 37
  4.1 Posterior distributions ................................ 37
  4.2 Implementation using MCMC .............................. 39
    4.2.1 Updating \( \mu \) using the Gibbs algorithm ............. 40
    4.2.2 Updating \( \lambda_y \) and \( \lambda_z \) using the Metropolis-Hastings algorithm ........................................ 40
4.2.3 Updating the vectors $\psi_x$, $\psi_y$, and $\tau$ .............................. 42
4.2.4 Updating $\tau_\psi$ and $\tau_r$ ................................................. 47
4.2.5 Updating $\mu_r$ and $\lambda_r$ .................................................. 48
4.3 Estimating the spatial process ..................................................... 49
  4.3.1 Sampling from the full conditional distribution of $z$ .......... 49
  4.3.2 Difficulties encountered when using the full conditional dis-
     tribution for $z$ ............................................................. 50
  4.3.3 Sampling from the distribution of $z'$ ............................... 51

5 Computing with semi-positive definite covariance matrices 55
  5.1 Background concerning positive matrices ............................. 55
  5.2 How the Cholesky algorithm works ..................................... 56
  5.3 A potential problem when sampling from the full conditional for $z$ .. 58
  5.4 A modified Cholesky algorithm for use with semipositive definite matrices 59
  5.5 Comparison between the new and old approaches with an example 62
  5.6 Using the modified Cholesky algorithm in practice ..................... 67
  5.7 Generating from the full conditional distribution for $z$ ............ 68
    5.7.1 Conditional distribution of $z_2$ given $z_1$ ........................ 69
    5.7.2 Full conditional distribution for $z_1$ .............................. 71
    5.7.3 Predicting $z'$ using $z_1$ ........................................... 73
  5.8 A simple example ............................................................. 75

6 Example: Application to the Piazza Road Superfund Site 78
  6.1 Introduction to the Piazza Road data set ............................. 78
  6.2 Analysis using a simple Gaussian covariance model ................... 81
6.2.1 Differences in implementation ........................................... 82
6.2.2 Results ................................................................. 84
6.3 Analysis using a process convolution approach ......................... 84
   6.3.1 An approach restricting all kernels to the same size ............. 84
   6.3.2 Full approach, allowing size of kernels to vary across locations 94

7 Conclusions and possible extensions ........................................ 103

Bibliography ............................................................................ 107

Biography ................................................................................ 111
List of Figures

1.1 Gaussian covariogram for various values of $\tau$ .................................. 5
1.2 Gaussian variogram for various values of $\tau$ .................................. 6
1.3 Exponential covariogram for various values of $\tau$ .............................. 7
1.4 Exponential variogram for various values of $\tau$ ................................ 7
1.5 Surface constructed using identical, isotropic kernels .......................... 10
1.6 Isotropic kernel used to generate Figure 1.5 .................................. 11
1.7 Surface constructed using identical, anisotropic kernels ....................... 12
1.8 Anisotropic kernel used to generate Figure 1.7 ................................. 13
1.9 Surface constructed using kernels varying in their orientations .............. 14
1.10 Surface constructed using kernels varying in their orientations and sizes . 15
3.1 A smooth, stationary random field obtained by taking a convolution of i.i.d. normal realizations, using the normal kernel pictured at top ................. 27
3.2 Ellipses derived from a sequence of foci points ($\psi_x, \psi_y$) ................. 31
5.1 After two iterations: (top) Results from the traditional algorithm; (bottom) Results from the modified algorithm ................................. 64
5.2 After three iterations: (top) Results from the traditional algorithm; (bottom) Results from the modified algorithm ................................. 64
5.3 After four iterations: (top) Results from the traditional algorithm; (bottom) Results from the modified algorithm ................................. 65
5.4 The completed decomposition returned by the modified algorithm ......... 66
5.5 The real trend surrounded by the sample data ................................. 75
5.6 The posterior means of the estimated spatial process .......................... 76
5.7 The posterior means of the estimated trend, together with the sample data
and the real trend ................................................................................. 77
6.1 Measured levels of contamination for chosen points at the Piazza Rd. site 80
6.2 Posterior mean surface using Gaussian covariance structure ................. 85
6.3 Posterior mean surface using changing orientation ellipses ................. 88
6.4 Orientation-varying ellipses averaged radially ................................. 90
6.5 Posterior mean surface using same-area ellipses ............................. 92
6.6 95% credible intervals for the fitted values plotted with observations (using
same-area ellipses) ........................................................................... 93
6.7 Posterior mean surface using ellipses that change in both orientation and
size ........................................................................................................ 96
6.8 Orientation and size-varying ellipses averaged radially .................... 97
6.9 Posterior mean surface using size-varying ellipses ........................... 99
6.10 95% credible intervals for the fitted values plotted with observations (using
size-varying ellipses) ......................................................................... 100
6.11 Location of the four points chosen for comparison ............................ 101
6.12 Posterior intervals for four chosen points at the Piazza Road site. ....... 102
7.1 Using disks to define a kernel .............................................................. 105
Chapter 1

Preliminaries

This dissertation presents approaches for modeling spatially correlated data; that is, data in which a variable(s) is correlated with another in a fashion that somehow depends on the locations of the two points. Such examples can be found in many areas, including, but not limited to, applications in environmental monitoring, geology, disease mapping, and image reconstruction. In some cases, the correlation is so weak that treating the values of the variable at different points as independent affects the analysis only mildly. However, in many cases, ignoring the spatial correlation inherent in a data set will result in misleading conclusions about the likely values of parameters, as well as a misunderstanding of the nature of the process which has produced the data.

Spatially correlated data can be found in many different examples and in many different forms. One possible situation is that in which we are faced with a variable or parameter can be viewed as a partial realization from a random process that is operating over the entire region of interest. A common case is that of a process that can be measured over a continuous index of space. A substantial amount of work has been done in this area, particularly in regards to geostatistics; this work can be motivated by the need to identify areas at which mining or drilling would be most
productive. In other examples, we may be interested in analyzing spatially correlated data that can only be measured at a finite collection of sites within the region of interest. Examples of such a collection of sites may be county seats, hospitals, or any other unit which aggregates and reports data from one location for an entire region.

In the case that one suspects spatial correlation, there are many possible models that can be used to facilitate analysis. Since the various approaches rely on different assumptions, it is important to identify these assumptions and determine which are appropriate for any given application. We try to address some of these concerns by explaining the terminology and mentioning common approaches to spatially correlated data sets.

Assume that we have a data set consisting of a series of locations $s$ and a set of measurements $z(s)$ taken at these locations. To begin a spatial analysis, the starting assumption is usually that the spatial process of interest is a realization of some random process $\{Z(s) : s \in D\}$ where $D$ is a subset of $d$-dimensional space (Cressie, 1993). A random process can be defined as a process that satisfies

$$F_{s_1,\ldots,s_m}(z_1,\ldots,z_m) = P(Z(s_1) \leq z_1, \ldots, Z(s_m) \leq z_m)$$

However, at this point, we still need to make some assumptions about stationarity in order to make inference feasible.

We begin with a brief discussion of variograms. A variogram can be constructed given that the following equation is true:

$$Var(Z(s_1) - Z(s_2)) = 2\gamma(s_1 - s_2) \quad (1.1)$$

This means that variance of the difference between any two points $s_1$ and $s_2$ is determined by $\gamma$, which is a function that depends only on the vector separating $s_1$
and \( s_2 \) (Cressie, 1993). Note that (1.1) can also be defined as

\[
Var(Z(s) - Z(s + h)) = 2\gamma(h)
\]

where \( h \) is the vector separating \( s_1 \) and \( s_2 \); this notation is may sometimes be more convenient.

The function \( 2\gamma(s_1 - s_2) \) defines the \textit{variogram} (Matheron, 1962). (The function \( \gamma(s_1 - s_2) \) is sometimes called the semivariogram.) This is often estimated from the data, and then used to determine the covariance structure that should be used to model the spatial process. Under the commonly-made assumption of constant mean over the region, written as

\[
E(Z(s)) = \mu
\]

we can re-write equation (1.1) as

\[
2\gamma(s_1 - s_2) = E((Z(s_1) - Z(s_2)^2)
\]

which expresses the variogram in terms of expected value. Examples of variograms are visible in Figures 1.2 and 1.4.

Note that by definition, the variogram must be zero when \( h \) is zero. However, due to small-scale variation, it is possible for there to be a discontinuity at the origin, such that \( \gamma(h) \to c \) as \( h \to 0 \). In this case, \( c \) is called the \textit{nugget effect} by Matheron (1962). This effect is often attributable to measurement error.

Another assumption that is often made about a spatial process is that of \textit{second-order stationarity} (also called weak stationarity). In addition to equations (1.2) and (1.1), second-order stationarity requires that the following condition be met:

\[
Cov(Z(s_1), Z(s_2)) = C(s_1 - s_2) \tag{1.3}
\]

Like \( \gamma \), the function \( C() \) can depend only on the vector separating \( s_1 \) and \( s_2 \), and is called the covariogram (Cressie, 1993). As was the case before, we can also write the
covariogram using the separation vector \( h \) as

\[
Cov(Z(s), Z(s + h)) = C(h)
\]

In the case that the function \( C() \) depends on \( h \) only through the length of the vector \( h \) (i.e. the orientation of \( h \) is not relevant), the covariogram is isotropic. In the case of weak stationarity, the variogram can be written

\[
2\gamma(h) = C(0) - C(h)
\]

Strict stationarity means that the probability function of \( Z \) for any two groups of points separated by a shifting vector \( h \) is the same, so that we have

\[
F(Z(x_{s_1}) \leq z_{s_1}, Z(x_{s_2}) \leq z_{s_2}, \ldots, Z(x_{s_n}) \leq z_{s_n}) = F(Z(x_{s_1} + h) \leq z_{s_1}, \ldots, Z(x_{s_2} + h) \leq z_{s_2}, \ldots, Z(x_{s_n} + h) \leq z_{s_n})
\]

Note that strict stationarity implies weak stationarity given that the second moment of \( F \) is finite (Cressie, 1993). This is true in the case of the Gaussian process, since it is completely defined by its mean and covariance.

In a situation in which isotropy can be stipulated, the Matérn class provides the form for a commonly used set of valid covariograms (Handcock and Stein, 1993) that can be written as functions of the \( \theta_1 \), a positive scaling parameter, and \( \theta_2 \), a positive parameter that controls the smoothness of the process.

\[
Cov(z_i, z_j) = \sigma^2 \frac{1}{2^{\theta_2 - 1} \Gamma(\theta_2)} \left( \frac{d_{ij}}{\theta_1} \right)^{\theta_2} B_{\theta_2} \left( \frac{d_{ij}}{2 \theta_1} \right)
\]

where \( B_{\theta_2} \) is the Bessel function of order \( \theta_2 \). This is quite a large class of isotropic models; the more commonly used come from the subclass

\[
Cov(z_i, z_j) = \sigma^2 \exp \left\{ -\frac{d_{ij}^\theta}{a} \right\}
\]
where $a > 0$ and $0 < b \leq 2$ (Wackernagel, 1998). Many variations exist and are commonly used. Two of the most common include the Gaussian (in the case where $\theta_2 \to \infty$) and exponential ($\theta_2 = \frac{1}{2}$) covariograms (Handcock and Stein, 1993). The marginal variance of each realization of $Z$ is given by $\sigma^2$; note that the marginal variance must be the same for each location $s$.

The Gaussian covariance structure is useful in many contexts, especially when it is desirable to fit a very smoothly varying spatial process. It is given by

$$Cov(z_i, z_j) = \sigma^2 \exp\left\{-\frac{d_{ij}^2}{\tau^2}\right\}$$

The scaling parameter $\tau$ controls how far apart points must be before the covariance between them is practically zero. For larger values of $\tau$, the spatial dependence reaches points further apart. In Figure 1.1, we see a Gaussian covariogram with several different values for $\tau$, allowing us to see how the differing values affect the co-
Figure 1.2: Gaussian variogram for various values of $\tau$

variogram. For comparison purposes, Figure 1.2 shows the corresponding variograms for each value of $\tau$.

The exponential covariogram is defined as

$$Cov(z_i, z_j) = \sigma^2 \exp\left(-\frac{d_{ij}}{\tau}\right)$$

Figure 1.3 shows exponential covariograms for the same values of $\tau$ featured in Figures 1.1 and 1.2. The plot shows that the covariance function does not change as smoothly with increasing distance as was the case with the Gaussian covariogram. Once again, the corresponding variograms are shown in Figure 1.4.

The range of procedures available for analyzing spatially correlated data sets has grown rapidly in recent years due mainly to an increase in the amount of computing power obtainable. The approaches are quite varied and may be tailored to particular types of data sets, with assumptions appropriate for these circumstances. We are
Figure 1.3: Exponential covariogram for various values of $\tau$

Figure 1.4: Exponential variogram for various values of $\tau$
particularly interested here in those that do not assume isotropy or do not assume stationarity.

For instance, one group of models that allow for anisotropy is given by Vecchia (1988). Assuming the that the spatial process can be represented as a Gaussian random field, this approach uses maximum likelihood estimation (obtained through an iterative algorithm to increase efficiency) to fit a covariance function for the data. However, the class of covariance models considered includes only those that with appropriate rotation and scaling can be transformed to isotropic processes. Although this approach is not restricted to case in which isotropy can be assumed, the covariance structure must be stationary and only certain types of anisotropy are permitted.

One interesting method for dealing with cases in which stationarity cannot safely be assumed is given by Sampson and Guttorp (1992). This approach is actually intended for use with temporal-spatial models, in which stationarity is assumed across time, but not across locations. The authors use multidimensional scaling techniques to transform the plane with the observations into another, in which the covariance structure is both isotropic and stationary. However, multiple observations (across time) are required at each location, in order to obtain point estimates of the covariance between locations.

Several frequentist kriging-based methods for spatially correlated data are proposed by Haas (1990) for uses in various situations. In particular, the proposed “moving window” algorithms allow for differing sample variograms at each location, which can account for non-stationarity in covariance structure and varying trend. These methods may have several disadvantages, including the difficulty of estimates of uncertainty, the assumption of isotropy in the “moving window” around each location, and in some cases, the assumption of the multivariate lognormal distribution for the spatial process.
Another approach which has been used successfully with very large, irregularly-spaced data sets is that of Nychka et al. (1999). The approach accommodates non-stationarity through the use of localized sets of basis functions. The authors make use of existing, but not commonly used, numerical algorithms to increase computational efficiency to allow the handling of very large data sets. However, one notable constraint of this model is the necessity of having the locations at which data was observed be a subset of the grid over which predictions will be made; this may be particularly inconvenient in the case of data sets in which the structure does not resemble a regular lattice of any sort.

Up to this point, we have spent a significant amount of time discussing the various forms of stationarity, and we have touched on some approaches which make use of this property. However, this dissertation proposes a method of dealing with spatial data that does not require an assumption of stationarity. We propose to use convolutions to handle these cases, and we describe this approach in detail in Chapter 3.

The basis for this approach can be found in several sources. In Silverman (1986), the author describes various methods to approximate the density function of an unknown univariate distribution using a probability density function as a kernel to smooth the observed data. Other mentions of the basics of this approach in the literature include generalizations from the moving average specification for a time series to a kernel convolution approach to spatially correlated data (Thiébaux and Pedder, 1987).

To summarize, we recall that any Gaussian process can be constructed by taking the convolution of a white noise process, so that the realization of \( Z \) at location \( s \) is given by

\[
Z(s) = \int_{\mathbb{R}^2} k(s-u)x(u) \, du
\]  

(1.4)

where \( x() \) is a white noise process and \( k() \) is the convolution kernel. A more complete
explanation is given in Chapter 3; see also Higdon (1998) and Higdon et al. (1999). The form of the convolution kernel is a critical choice in obtaining the desired $z$.

![Figure 1.5: Surface constructed using identical, isotropic kernels](image)

Here we have chosen to use the bivariate normal distribution as our convolution kernel, primarily for the convenience of interpretation and implementation. To allow for non-stationarity, the kernels must be allowed to change at the various locations $s$; if the spatial process is actually stationary, they will all be the same. The problem of choosing the parameters of the kernels is a major focus of this work. The following figures show the choice of kernels affects the realizations of $Z$. To find the realization of the spatial process $Z$ at each of the $n$ points $s_1, \ldots, s_n$, we use the formula

$$z_i = \sum_{j=1}^{n} k(s_i - s_j)x_j$$

where $k()$ is a bivariate normal distribution and the $x_i$ is a realization from a white
Figure 1.6: Isotropic kernel used to generate Figure 1.5

noise process. This is just the discrete analogue of equation (1.4).

In Figures 1.5 and 1.7, we see realizations of \( Z \) obtained after using the same bivariate normal distribution at each location \( s_i \). In Figure 1.5, we use an isotropic bivariate normal distribution. Figure 1.7 displays the results that are obtained using an anisotropic kernel. We use ellipses to delineate the one standard deviation boundaries of each bivariate normal kernel. The kernel used to make Figure 1.5 is shown in 1.6 so that we can see it on the same scale as Figure 1.5, together with the ellipse which we used to parameterize it. Likewise, Figure 1.8 gives the kernel utilized in making Figure 1.7. In Figures 1.5 and 1.7, we see the same ellipses superimposed over the image at each point for which they are used, although each has been scaled down to make a clear display. We can see that although these two plots both portray a stationarity process, the choice of kernels has a strong influence over the realization
Figure 1.7: Surface constructed using identical, anisotropic kernels of $Z$ that is obtained.

Now we take a look at two examples with non-stationary covariance structure. Figure 1.9 shows the realization obtained when using kernels that are allowed to vary in their orientation. These kernels were constructed using their ellipse analogues. The $x$ and $y$ coordinates ($\psi_x, \psi_y$) of a focus point for each of the $n$ ellipses were drawn from the following distributions:

$$
\psi_x \sim N(0, \Sigma_\psi) \\
\psi_y \sim N(0, \Sigma_\psi)
$$

where each covariance matrix element $\Sigma_\psi$ is defined according to the Gaussian covariance structure previously discussed with scale parameter $\tau = 3$. This construction forces the ellipses to change smoothly over the area, but still allows for a sizeable amount of variation.
Figure 1.8: Anisotropic kernel used to generate Figure 1.7
Figure 1.9: Surface constructed using kernels varying in their orientations

The final example, shown in Figure 1.10, allows for the most freedom in choosing the bivariate normal kernels. In addition to incorporating the changes in orientation given by the $\psi$s in Figure 1.9, we also include a scaling variable $\tau$ that serves to shrink or expand the area of the kernels. The case of the $\tau$ vector is similar to that of the $\psi$ vectors; $\tau$ is a realization of

$$\tau \sim N(1, \frac{1}{\lambda_r} \Sigma_r)$$

where each covariance matrix element is given by the product of the marginal variance, $\frac{1}{\lambda_r} = \frac{1}{11}$, and the corresponding element of the Gaussian correlogram, $\Sigma_r$, with scale parameter $\tau = 50$. This allows each ellipse to differ from its neighbor in both size and orientation. However, since the scale factor in the covariogram for $\tau$ is larger than that for $\psi_x$ and $\psi_y$, the sizes of the ellipses are forced to vary more smoothly
over the region of interest.

![Surface constructed using kernels varying in their orientations and sizes](image)

**Figure 1.10:** Surface constructed using kernels varying in their orientations and sizes

Using the last four examples, we can see that the choice of convolution kernel makes a large difference in the spatial process that is obtained when these kernels are convolved with a white noise process. This dissertation focuses on how to find these kernels and use them to build an appropriate covariance structure for the data. In Chapter 2, we discuss a general spatial model and the parameters that govern it. Chapter 3 address the issue of the convolution kernels in more detail, discussing the theory, the parameters governing the kernels, and the prior distributions that are necessary for a Bayesian approach. The implementation of Markov Chain Monte Carlo and computational considerations are addressed in Chapter 4. Computational issues of particular consideration include round-off error and its potentially deleterious effect on matrix operations, which are discussed in Chapter 5. For instance,
one such aspect is the ability of round-off error to make positive definite matrices stored in the computer effectively semipositive definite. In Chapter 6, we explore the application of this methodology in an example involving waste remediation at an EPA Superfund site. Finally, we end with conclusions and discussion about possible directions for future research.
Chapter 2

Introduction of the model

2.1 A general model

One fairly simple way to model a process with a spatial component is to think about the data as a combination of three parts, according to the following equation

\[ y = X\beta + Dz + \varepsilon \]  \hspace{1cm} (2.1)

This form is quite common and is reminiscent of spatial models used by many other authors, including Higdon et al. (1999), Besag and Higdon (1999), and Vecchia (1988). The \( n \) observed values are denoted by \( y \), which is in the form of a \( n \times 1 \) vector. The term \( X\beta \) gives the trend for the model, and its presence allows us to take into account information that we may have about the region of interest that is not taken into account as a part of the spatial process. \( X \) represents a design matrix of \( n \times p \) known covariates, while \( \beta \) is a \( p \times 1 \) vector of parameters to be fitted. In some data sets, we may not have covariates available to help us distinguish changes in the trend. In those cases, it is useful to think of the trend as being constant over the whole region. We deal with this situation by restricting \( \beta \) to length 1, with \( X \) a \( n \times 1 \) vector of ones. If this is the case, we simplify the notation by denoting \( \beta \) as \( \mu \),
which is a more obvious way of denoting the mean, yielding

\[ y = \mu \mathbf{1} + D z + \varepsilon \]

The second term, through which we will focus the majority of our efforts, gives the spatial component inherent in the process. In particular, each element \( z_i \) of the \( m \times 1 \) vector \( z \) gives the value of the spatial component for observed site \( i \). Here, the length of \( z \) must be equal to the number of sites \( m \) at which observations were taken, since the spatial process has a value for each location. However, it is possible that more than one observation has been made at some or all of the \( m \) sites, so that \( n > m \). The \( n \times m \) matrix \( D \) serves to indicate at which site a given observation was taken; if \( n = m \), \( D \) is identical to the \( n \times n \) identity matrix. We assume that there is some white noise in addition to the trend and spatial components, perhaps due to measurement error, experimental error, or other sources. This noise component is given by the vector \( \varepsilon \).

Given this model, we consider the likelihood for the observed data (denoted by \( y \)). Since we have already assumed that the measurements contain some element of white noise \( \varepsilon \), it seems logical to hypothesize that \( y \) follows a multivariate normal distribution given below.

\[ y \sim N(X \beta + D z, \frac{1}{\lambda_y} \mathbf{I}) \quad (2.2) \]

Here, \( \mathbf{I} \) is the identity matrix, which we use in conjunction with \( \frac{1}{\lambda_y} \) to remind us that the variance of the white noise process is the same for each observation and that the elements of \( \varepsilon \) are independent.

The likelihood for \( y \) contains several terms that are unknown, for which we may or may not have any knowledge of their approximate value or range. The Bayesian approach allows us to specify prior distributions for these parameters which convey any information that we already have about the likely value of the parameter. To
complete our understanding of the model, we need to determine these priors, and from them, the full conditional distributions of each of the parameters.

2.2 Priors

In the Bayesian paradigm, unknown parameters are viewed as random variables. We attempt to quantify our knowledge about a parameter, including the values it is most likely to have, whether it is restricted to positive values, and other information. This differs from other, frequentist approaches, because it allows us to incorporate what we already know about an unknown from other experiments, outside study, or logical reasoning. If we have already obtained a significant amount of knowledge about a parameter, the variance in its prior distribution will be reduced accordingly. If we are still very unsure about the parameter, the variance will be large. Thus, the choice of prior distributions plays an important role in our analysis, and we discuss these choices in relation to our model in this section.

First, consider the vector $\beta$, which denotes the coefficients for the covariates given in $X$. A variety of priors could be appropriate, depending on whether the $\beta$s are exchangeable, independent, etc. Choice of prior may depend substantially on knowledge about the types of measurements recorded as a part of $X$. However, as mentioned in the previous section, a common and simple case occurs when no covariate data is available. In this situation, it may be appropriate to take $\beta$ as a vector with a length of just one, and $X$ as a $n \times 1$ matrix consisting of only ones. In this case the $X\beta$ term would represent the mean of the observed variable $y$ over the whole region, and could be more simply represented as $\mu 1$. If there is not a lot of information available about the likely value of $\mu$, a non-informative prior may be a convenient choice. Then, the prior for $\mu$ can be written as

$$\mu \sim U(-\infty, \infty)$$
This, combined with the likelihood given in (2.2) gives the full conditional for $\mu$ as

$$
\mu | y, \lambda_y, \lambda_z, \Sigma \sim N \left( \frac{1}{n} (y - Dz)^T 1, \frac{1}{n\lambda_y} \right)
$$

Although the white noise parameter $\varepsilon$ has mean zero, we examine its variance. As mentioned before, we have assumed the $\varepsilon_i$s to be independent and identically distributed with common mean, 0, and precision, $\lambda_y$. We need to set a prior for $\lambda_y$. A convenient choice of prior for a precision parameter is the gamma distribution, since this is the conjugate prior. We postpone the choice of the parameters to govern the prior for right now in order to focus on the form of the prior and the full conditional distributions of $\lambda_y$. We can write the prior as

$$
\lambda_y \sim \Gamma(a_{\lambda_y}, b_{\lambda_y})
$$

where the gamma distribution is given by

$$
\pi(\lambda_y | a_{\lambda_y}, b_{\lambda_y}) = \frac{b_{\lambda_y}^{a_{\lambda_y}} \lambda_y^{a_{\lambda_y} - 1} exp(-b_{\lambda_y} \lambda_y)}{\Gamma(a_{\lambda_y})}
$$

From here, we can see the full conditional distribution for $\lambda_y$ can be written

$$
\pi(\lambda_y | z, y, \mu, a_{\lambda_y}, b_{\lambda_y}) \sim \Gamma \left( \frac{n}{2} + a_{\lambda_y}, b_{\lambda_y} + \frac{1}{2} (y - (\mu 1 + Dz))^T (y - (\mu 1 + Dz)) \right)
$$

Now we consider the possible choices for the parameters of the prior distribution, $a_{\lambda_y}$ and $b_{\lambda_y}$. If we are very uncertain about what kind of variance our white noise process has, setting the parameters so that the distribution is very flat is an option. For instance, setting $a_{\lambda_y} = 1.0$ and $b_{\lambda_y} = 0.005$ yields a mean of 200 and a variance of 40000. In many cases, information about the process determining $y$ can help determine what the scale of the error is likely to be, in which case a more informative prior can be chosen. Depending on the values of the other parameters, especially those
governing the spatial process $z$, it is possible that giving a more informed prior can help differentiate more easily the variation that is due to noise and the variation that is attributable to the spatial process.

Consider the parameter $z$, which represents the value of the spatial field at locations at which observations have been made. We hypothesize that this is a Gaussian process, and that its mean is zero, since the overall trend has already been accounted for in the model by $X \beta$. We represent the covariance matrix for $z$ by $\frac{1}{\lambda_z} \Sigma$, where $\frac{1}{\lambda_z}$ gives the marginal variance for each $z_i$ and each element $\Sigma_{ij}$ of $\Sigma$ gives the correlation between the spatial process at site $i$ and $j$. In hierarchical fashion, $\Sigma$ depends on its own parameters; what these are depends on the covariance structure that is used to model the parameter $z$. There are many possibilities for modeling the covariance structure of a spatial process; in particular, we focus on a process convolution approach in later chapters.

Given these parameters, it is reasonable to write the prior for $z$ as

$$z|\lambda_z, \Sigma \sim N(0, \frac{1}{\lambda_z} \Sigma) \quad (2.3)$$

With reference to the likelihood, the full conditional distribution for $z$ can then be written as

$$z|\Sigma, \lambda_y, \mu, y, \lambda_z \sim N(\lambda_y (\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1} D^T(y - \mu 1),$$

$$(\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1}) \quad (2.4)$$

Since $\Sigma$ can be determined using a number of different approaches, the parameters which can play a part in it must be discussed individually with each different model.

As mentioned previously, $\frac{1}{\lambda_z}$ represents the variance of the marginal distribution of any element $z_i$ of the spatial process. Since $\lambda_z$ is a precision parameter, a convenient choice of prior is one from the gamma distribution; this is very similar to the case of
\( \lambda_y \). Once again, it may be hard to assess what prior knowledge, if any, is available to help determine the parameters of the prior distribution for \( \lambda_z \), which are denoted as \( a_{\lambda_z} \) and \( b_{\lambda_z} \). If there is very little prior information, it may again be wise to choose the parameters \( a_{\lambda_z} = 1 \) and \( b_{\lambda_z} = 0.005 \) in order to obtain a prior distribution with large variance. As in the case of \( \lambda_y \) we obtain the prior distribution

\[
\lambda_z | a_{\lambda_z}, b_{\lambda_z} \sim \Gamma(a_{\lambda_z}, b_{\lambda_z})
\]

The full conditional distribution for \( \lambda_z \) is then given by

\[
\lambda_z | z, \Sigma \sim \Gamma(a_{\lambda_z} + \frac{n}{2}, b_{\lambda_z} + \frac{1}{2}z^T\Sigma^{-1}z)
\]

Glancing through the previous paragraphs, we notice that depending on the number of sites, we may need to update a large number of parameters in each iteration of our MCMC scheme. Recall, for instance, that the length of the vector \( z \) is the number of sites \( m \) at which observations have been taken, and that therefore the covariance matrix \( \Sigma \) has dimension \( m \times m \). This means that we would need to update a potentially large vector \( z \) with every iteration, inverting the matrix \( \Sigma \) each time. Inversion of a matrix is a computationally intensive process which is very sensitive to numerical problems, particularly when the correlation between the spatial process at some or all sites is large. In order to save computational time and resources, it is worthwhile to consider trying to avoid updating the \( z \) vector with every iteration in the MCMC algorithm. One option is to integrate the \( z \) vector out of the joint distribution of parameters, which is

\[
\pi(z, \mu, \lambda_y, \lambda_z, y, \Sigma) = L(y | z, \mu, \lambda_y) \pi(z | \lambda_z, \Sigma) \pi(\mu) \pi(\lambda_y) \pi(\lambda_z) \pi(\Sigma)
\]  

(2.5)

If \( z \) is no longer part of the update scheme for the other parameters, we do not need to include an update step for \( z \) at every iteration. Instead, we are free to include it only
when we feel that the other parameters have converged to their true distributions, and we want to get an idea of what the spatial process is like given these other parameters.

Upon substituting the appropriate priors into (2.5), we want to focus on terms containing the vector $z$; the rest of the terms will not be affected by the integration. We then integrate over all possible values of the vector and organize the remainder, focusing on the terms containing $y$. Then, the likelihood for $y$ (without the $z$ term) must be

$$L(y|\mu, \lambda_y, \lambda_z, \theta) \propto |S|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(y - \mu 1)^T S (y - \mu 1)\right) \quad (2.6)$$

where

$$S = \lambda_y [I - D(\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1} \lambda_y D^T] \quad (2.7)$$

We recognize (2.6) as a multivariate normal distribution with mean $\mu 1$ and precision matrix $S$. It is often more intuitive to think in terms of the variance for (2.6), rather than the precision. To get the variance for $y$, we can add the variances for $Dz$ and $\varepsilon$ (since these are independent). The variance of $z$ is given by $\frac{1}{\lambda_z} \Sigma$; when $z$ is multiplied by a constant (in this case $D$), the variance becomes $D \frac{1}{\lambda_z} \Sigma^{-1} D^T$. When added to the variance of $\varepsilon$, which is $\frac{1}{\lambda_y} I$, we get

$$\text{Var}(y) = S^{-1} = \frac{1}{\lambda_z} D \Sigma^{-1} D^T + \frac{1}{\lambda_y} I$$

However, we can still use the $z$ parameter to guide our intuition. For instance, if we want to know the values of the $z$ parameters, we can still use a Gibbs sampling step to generate from the full conditional of $z$, which is given in (2.5), as we iterate through our MCMC routine. Since the other parameters will no longer depend on $z$, it may then be reasonable to only deal with $z$ only after the end of the burn-in period, and then not at every iteration. This may facilitate explanation of the results.
by allowing us to still give a posterior mean and variance for the spatial process at the various locations, and may help us make predictions for locations at which no observations are available.

Now that we have a new version of the likelihood function, we can re-calculate the full conditionals accordingly, obtaining

\[
\pi(\mu|y, \lambda_y, \lambda_z, \Sigma) \sim N((1^T S)^{-1} 1^T S y, (1^T S 1)^{-1})
\]

\[
\pi(\lambda_y|y, \mu, \lambda_z, \Sigma) \propto |S|^{\frac{1}{2}} \lambda_y^{a_y-1} \exp(-\frac{1}{2} (y - \mu 1)^T S (y - \mu 1) - b_{\lambda_y} \lambda_y)
\]

\[
\pi(\lambda_z|y, \mu, \lambda_y, \Sigma) \propto |S|^{\frac{1}{2}} \lambda_z^{a_z-1} \exp(-\frac{1}{2} (y - \mu 1)^T S (y - \mu 1) - b_{\lambda_z} \lambda_z)
\]

Refering to (2.7), note that $S$ depends on $\lambda_y$, $\lambda_z$, and $\Sigma$. This will add some computational complexity to the MCMC update scheme, since $\Sigma$ will have to be re-calculated with each change in the aforementioned parameters. Also, only the parameter $\mu$ has a full conditional distribution in closed form, which will change our MCMC updating scheme to contain more Metropolis-Hastings steps than it did previously; we can expect some costs in terms of computing power for this reason also.
Chapter 3

The process convolution approach

3.1 Motivation

A difficulty with methods commonly in use is that many of them make assumptions about covariance structure which may not always be appropriate. For instance, as we discussed in Chapter 1, many assume varying degrees of stationarity. The Matern class covariance functions mentioned before assume isotropy, meaning that the covariance between any two points only depends on the Euclidean distance between them. In many spatial processes it is unreasonable to make either of these assumptions; for instance, natural forces such as weather, landscape features, and other large-scale processes may affect the various areas of the region in question differently. If this is the case, then an approach which only takes into account how far points of observation are from one another, and not the distance and direction which separate these points, will miss some peculiarities of the spatial dependence structure.

The process convolution approach was developed in order to deal with cases in which stationarity and isotropy cannot be safely assumed. This methodology allows the position of points of observation to be taken into account, and allows us to see any patterns that are particular to a certain subset of the region in question.
3.2 Process convolution approach

In this section, we discuss the formulation of a “process convolution” approach to modeling non-stationary covariance. To begin, we limit ourselves to the stationary case, addressing the development of the approach in simpler cases. The second section then develops this approach further for use in the case of non-stationarity.

3.2.1 The approach in the case of stationarity

Figure 3.1 shows a simple example in one dimension, in which we have used the convolution approach to obtain a smooth, stationary Gaussian field from i.i.d Gaussian random variables. The vertical lines represent independent realizations from a normal distribution; the bold line represents the smooth field \( z \) that is generated using

\[
z_i = \sum_{j=1}^{n} k(s_i - s_j)x_i
\]  

(3.1)

where \( z_i \) is the value of the smoothed process at point \( s_i \). The smoothing kernel \( k \) that was convolved with these Gaussian realizations is the normal distribution pictured at the top of Figure 3.1. To obtain \( z_i \), we first multiply the realization of the i.i.d. Gaussian random variables given as \( x_j \) for each point \( s_j \) by the height of the the kernel, centered at \( s_i \), at location \( x_j \). The sum of these values gives us \( z_i \). This is equivalent to the form of a moving average time series process given in equation (3.2), in which the value of the time series at time \( t \) is given as a linear filter with white noise innovation. Stationarity of \( z \) can be assured in this case without knowing the exact form of the \( b \) terms, since \( \varepsilon \) represents a white-noise process with finite variance (Diggle, 1993).

\[
z_t = \sum_{k=0}^{\infty} b_k \varepsilon(t - k)
\]
Figure 3.1: A smooth, stationary random field obtained by taking a convolution of i.i.d. normal realizations, using the normal kernel pictured at top.
We begin by examining a simpler case in which our spatial process, previously denoted \( z \), is stationary. We expand the discrete time series moving average model in 3.2 to the continuous case of spatial coordinates (Thiébaux and Pedder, 1987) in more than one dimension. In this case, we have an integral, rather than a sum.

\[
    z(s) = \int_{\mathbb{R}^2} k(s - u) x(u) \, du
\]

(3.2)

The convolution kernel \( k \) is a function that provides the weights corresponding to the series \( b \) in equation 3.2. The white-noise innovation process corresponds to \( x(u) \), which is the value of the Gaussian white noise process at point \( u \). Note that this is the continuous version of the summation used to create Figure 3.1, given in 3.1. In the stationary case, this kernel \( k(s) \) must be the same for every location \( s \).

In order to obtain the correlation matrix for our spatial process \( z \), we need to be able to find the correlation between any two points separated by a given distance \( d \). We then build the correlation matrix for \( z \), called \( \Sigma \), entry by entry using the correlation function

\[
    \rho(d) = \int_{\mathbb{R}^2} k(s) k(s - d) \, ds
\]

Consider the standard normal kernel in two dimensions. It can be expressed as

\[
    k(s) = \frac{1}{2\pi} \exp -\frac{1}{2} s^T s.
\]

Notice that it is isotropic, so that the expression only depends on the distance between \( s_1 \) and \( s_2 \). If we want to allow for a change in size of the axes or rotation, we can add the covariance matrix \( \Sigma \), and define

\[
    k(s) = \frac{1}{2\pi} \Sigma^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} s^T \Sigma^{-1} s \right\}
\]

If we define \( s' = \Sigma^{-\frac{1}{2}} s \), then \( k(s) \) is isotropic on the transformed coordinate axes that have been defined by \( \Sigma \). However, since the kernel is the same at every site in the region, the process \( z \) must still be stationary.
3.2.2 The process convolution approach in the case of non-stationarity

In order to allow for non-stationarity, the model has to allow different kernels to be used at all sites within the region. So, we need \( m \) kernels for \( m \) different sites, and we denote as \( k_s() \) the kernel associated with site \( s \). This kernel is centered at the point \( s \). Now, to get the correlation between points \( s \) and \( s' \), we write

\[
\rho(s, s') = \int_{\mathbb{R}^3} k_s(u)k_{s'}(u) \, du
\]  

(3.3)

Since each kernel is now slightly different, it is important to think carefully about the form of each and to consider how the correlations between the process at site \( i \) and site \( j \) could be calculated. We denote the covariance matrix \( \Sigma \) for the kernel at point \( s \) as \( \Sigma(s) \) and that for the kernel at point \( s' \) as \( \Sigma(s') \). Now we can further specify these in the form (Higdon et al., 1999)

\[
\Sigma(s) = \begin{pmatrix}
    a^2 & \rho ab \\
    \rho ab & b^2
\end{pmatrix}
\quad \text{and} \quad
\Sigma(s') = \begin{pmatrix}
    a'^2 & \rho'a' b' \\
    \rho'a' b' & b'^2
\end{pmatrix}
\]

Now using the formula given in (3.3), we integrate over all values of \( s \) and \( s' \). Although this integration is mathematically complex, we arrive at an answer which can be written most easily if it is viewed in parts.

We write the correlation between any two points \( s \) and \( s' \) as a simple formula dependent on three variables, \( q_1, q_2, \) and \( W \) (Higdon et al., 1999).

\[
\rho(s, s') = \frac{1}{q_1} \exp\left\{-\frac{1}{q_2}(s-s')W(s-s')\right\}
\]  

(3.4)

Then we define each of the constituent parts as follows:

\[
W = \begin{pmatrix}
    b^2 + b'^2 & -\left(\rho ab + \rho'a' b'\right) \\
    -\left(\rho ab + \rho'a' b'\right) & a^2 + a'^2
\end{pmatrix}
\]
\[ q_1 = 2\pi a a' b b' \sqrt{(1 - \rho^2)(1 - \rho'^2)} \sqrt{\frac{(\rho^2 - 1)b^2 + (\rho'^2 - 1)b'^2}{(\rho^2 - 1)(\rho'^2 - 1)b^2 b'^2}} \]

\[ \times \sqrt{\frac{2\rho \rho' a a' b b' + a^2((\rho^2 - 1)b^2 - b'^2) + a'^2((\rho'^2 - 1)b'^2 - b^2)}{a^2 a'^2((\rho^2 - 1)b^2 + (\rho'^2 - 1)b'^2)}} \]

\[ q_2 = 2(-2\rho \rho' a a' b b' - a^2((\rho^2 - 1)b^2 - b'^2) - a'^2((\rho'^2 - 1)b'^2 - b^2)) \]

Once we have determined how to calculate the correlation between any two points in the region, it is reasonable to ask how we can parameterize the matrix \( \Sigma \). It would be convenient to have a methodology that can be understood on a more intuitive level than what has just been presented. On the mathematical side, the parameterization must be such that prior distributions can be easily assigned to parameters and such that the kernels change smoothly over the region.

### 3.3 Connection of process convolution idea to the visualization of an ellipse

Visualize a bivariate normal distribution in three dimensional space. If we approach the distribution from above (from the positive z-direction), we can see that any xy-plane that has constant \( z \) and intercepts the bivariate normal distribution at any point except the very top point of the curve will form an ellipse in the piece of the bivariate kernel that it cuts. So, this allows us to use the equation of an ellipse and the visualization of an ellipse as one way to more intuitively parameterize \( \Sigma \), as in Higdon et al. (1999) and Higdon (1998).

So, our job is to now define a set of ellipses that varies smoothly over the region. We can think of defining an ellipse through a set of three parameters: its center, one of two focus points, and its area. Since each ellipse represents a kernel that is centered over a certain location \( s \), we know that the center of the ellipse is also
the point $s$. We keep the areas of the ellipses set at a set value, which we denote $A$. (Another parameter, $\tau$, exists to allow the kernels to increase their range of influence.) In our case, we have set the areas of the ellipses at 3.5, which was chosen after some experimentation, but can be suitably adjusted as necessary. In order to allow the ellipses to vary, we can then manipulate only the set of coordinates for a focus point $(\psi_x, \psi_y)$ and a scaling factor $\tau$ for each ellipse. Figure 3.2 depicts the shape and orientation of various ellipses, derived from a sequence of foci points $(\psi_x, \psi_y)$. The $\psi_x$s and the $\psi_y$s are drawn from separate, independent Gaussian processes with covariogram $\exp\{-d^2/10^2\}$, where $d$ denotes distance. For each focus $(\psi_x, \psi_y)$, the resulting ellipse is plotted directly below it.

To help us find $\Sigma$ in terms of these parameters, we use some geometric properties of ellipses; in particular, we are interested in the formulas to calculate the area of an
ellipse and the major and minor axes of an ellipse. These can be written as

\[ A = \pi ab \]
\[ \|\psi\|^2 = a^2 + b^2 \]

Remember that \( a \) is traditionally defined as the length of the major axis and \( b \) as the length of the minor axis. Also, the distance from the center of the ellipse to either of the two focus points is represented here by \( \|\psi\| \). Once we have know the area \( A \) and calculate \( \|\psi\| \), we can calculate the Gaussian kernel whose covariance matrix, \( \Sigma \), is given by the following formula:

\[
\Sigma^{\frac{1}{2}} = \tau \begin{pmatrix}
\sqrt{\frac{4A^2 + \|\psi\|^2}{2\pi}} + \frac{\|\psi\|^2}{2} & 0 \\
0 & \sqrt{\frac{4A^2 + \|\psi\|^2}{2\pi}} - \frac{\|\psi\|^2}{2}
\end{pmatrix}^{\frac{1}{2}}\begin{pmatrix}
\cos \alpha & \sin \alpha \\
-sin \alpha & \cos \alpha
\end{pmatrix} \tag{3.5}
\]

where \( \alpha = \tan^{-1} \frac{\psi_y}{\psi_x} \). So, this equation provides a correspondence between ellipses, defined by their centers, foci, and area, and the bivariate Gaussian kernels that they represent.

Note that since each ellipse can have different orientation and size, we have a vector of parameters with the same length as we have number of observation sites. We denote these vectors \( \psi_x, \psi_y \), and \( \tau \). To force these parameters to change slowly over space and for computational convenience, we use multivariate normal prior distributions on each one. The details involved in determining the prior distributions and related issues are discussed at length in the next section.

### 3.4 Priors for process convolution specific parameters

In accordance with our Bayesian methodology, we must determine prior distributions for the parameters introduced in the last section. It seems logical that although the
orientation and size of the ellipses for each data point will likely be different, sites close to one another in space should be similar in their orientation and size. This idea will help us to determine the prior distribution for the vectors $\psi_x$, $\psi_y$, and $\tau$.

### 3.4.1 Priors for foci coordinates $\psi_x$ and $\psi_y$

First, we consider the prior distributions for $\psi_x$ and $\psi_y$. Since we want the ellipses to vary smoothly over the region, we need a distribution which can incorporate some measure of spatial dependence between the foci of the various kernels. The most convenient choice is to use a multivariate normal distribution prior for both $\psi_x$ and $\psi_y$. We choose a simple Gaussian covariance structure both for computational convenience and to ensure that the ellipses change smoothly. However, other choices could suffice, depending on prior knowledge and on the degree of smoothness desired. The covariance matrices for $\psi_x$ and $\psi_y$ are assumed to be the same, since there is no reason to expect that the spatial dependence will be stronger or weaker for the x-coordinates versus the y-coordinates of the foci.

For the mean vector of both prior distributions, we use a vector of all zeroes. If both $\psi_x$ and $\psi_y$ consist of all zeroes, then the foci for each ellipse will actually fall at the ellipse’s center. So, in this case, we have circles (recall that a circle is a special case of an ellipse in which the foci have degenerated to the center), meaning that the local dependence is the same in all directions and in all areas of the field (isotropy). Since we often have very little prior knowledge about the spatial dependence structure in the region, including whether or not we non-stationarity is even present, it seems appropriate to a priori assume isotropy, and allow the data to determine the extent and nature of the non-stationarity.

The Gaussian covariance structure for $\psi_x$ and $\psi_y$ depends on the specification of a scale parameter. We allow the data to help determine the scale of the spatial
dependence by including the hyperparameter $\tau_{\psi}$ in the formulation of the correlation matrix. We denote this matrix as $\Sigma_{\tau_{\psi}}$, with each element in row $i$ and column $j$ given by

$$
Corr(\psi_x(i), \psi_x(j)) = Corr(\psi_y(i), \psi_y(j)) = \exp\left(-\frac{d_{ij}^2}{\tau_{\psi}}\right)
$$

(3.6)

Recall that $d_{ij}$ represents the Euclidean distance between the points $i$ and $j$. The Gaussian covariance structure also should include a parameter that gives the marginal variance for any element of the vector. In this case, our prior specification for $\psi_x$ and $\psi_y$ assumes that this value is simply one, so that the covariance matrix is identical to the correlation matrix. We do not want the foci points to range far from zero. This would lead to very eccentric ellipses, while we believe that more weight in the prior should be placed on isotropy, rather than extreme directional dependence. Unless the data specifies otherwise, we would rather that the ellipses tend toward circular shapes, which are given by foci very close to or on the origin. Summarizing these parameter choices, we have the prior distributions

$$
\psi_x \sim N(0, \Sigma_{\tau_{\psi}})
$$

$$
\psi_y \sim N(0, \Sigma_{\tau_{\psi}})
$$

The covariance matrix for $\psi_x$ and $\psi_y$ is governed by a scale parameter, $\tau_{\psi}$, through equation (3.6). Due to this role, by definition, $\tau_{\psi}$ must be positive. It seems logical to use the gamma distribution as a prior in this case. We choose the parameters $\alpha_{\tau_{\psi}} = 1.5625$ and $\beta_{\tau_{\psi}} = 0.0625$, since these place higher probability on a range of values that would help the kernels to vary smoothly. This is still a large range (the variance for $\tau_{\psi}$ is 400), so the structure will still be largely determined by the data.

$$
\tau_{\psi} \sim \Gamma(\alpha_{\tau_{\psi}}, \beta_{\tau_{\psi}})
$$
3.4.2 Priors for coordinate axis shrinking/expansion $\tau$

Now, we take a look at the prior distribution for $\tau$. Recall that each $\tau_i$ is a (positive) factor that controls the shrinking/expansion of the axes relative to kernel $i$. Since we want to express the dependence among the elements of $\tau$, another multivariate normal is a simple choice. The parameters for this distribution are given as mean $\mu_\tau$ and covariance matrix given by $\frac{1}{\lambda_\tau} \Sigma_\tau$. The reason for this choice is that we believe the spatial dependency between the $\tau$ elements is likely to be different than that for the $\psi_x$ and $\psi_y$ vectors. In most cases, the shrinking or stretching of the coordinate axis is thought to change even more smoothly over the region than the shape and orientation of the ellipses, in which case $\tau$ may be larger than $\tau_\psi$. Even though the elements of $\tau$ must be positive, this distribution has negligible density in the region where the $\tau$ elements are non-positive, so this is not likely to pose a significant problem. Then, for the prior for $\tau$, we have

$$
\tau \sim N(\mu_\tau \mathbf{1}, \frac{1}{\lambda_\tau} \Sigma_\tau)
$$

In this case, we assume that the mean vector for $\tau$ is the same for each element $\tau_i$. If we had strong prior information about what the value for $\tau$ should be at various points across the region, this algorithm could be implemented with a $\mu_\tau$ vector with differing elements. In most situations, though, our initial belief is that all elements of $\tau$ are centered around the same mean, giving us the mean vector $\mu_\tau \mathbf{1}$ for our prior on $\tau$. If desired, we can also put a prior distribution on $\mu_\tau$. One possibility would then be to use a normal prior for $\mu_\tau$ of the form

$$
\mu_\tau \sim N(m, s^2)
$$

where the mean $m$ and the variance $s^2$ are chosen appropriately for the problem. In our case, these were $m = 15$ and $s = 2.5$. 

35
We introduce the parameter $\tau_r$ to control the correlation matrix for $\tau$ in the same way that $\tau_{\psi}$ controls the correlation matrix for $\psi_x$ and $\psi_y$. Of course, this parameter must also be positive. Further, in most applications we believe that it will be larger than $\tau_{\psi}$, so as to force the ellipses to change size smoothly over the region. If the ellipses’ sizes are allowed to change too quickly, we are likely to encounter a situation in which the spatial process cannot be easily separated from the noise. This would result in a situation in which the spatial process is seen to “chase the data”, or follow the data too closely. To prevent this, we choose a relatively large value for $\tau_r$, 50, which allows for a large dependence between neighboring locations. (Refer to Figure 1.1 to see how the choice of scaling parameter affects the correlation of the spatial process between points separated by various distances.)

The parameter $\lambda_r$ represents the marginal precision of each element of $\tau$. Unlike the case of $\psi_x$ and $\psi_y$, there is not reason to believe that the marginal variance of the elements $\tau$ is so constrained. Instead, we would like to allow some individual variation among the elements of $\tau$. For this reason, we choose the value $\lambda_r = 0.1$, but other values can be chosen appropriately for the application.
Chapter 4

Posterior exploration

In Chapter 2, we discussed a general model for spatial data, which would allow us some flexibility in determining which covariance structure would be most appropriate for a given problem. Chapter 3 discusses one possible approach to modeling covariance structure; in particular, we propose a form which can be used in the case of non-stationarity. This chapter will treat the topics of posterior distributions, MCMC implementation, and estimation of the spatial process at unobserved points.

4.1 Posterior distributions

First, we review some key points from Chapter 2. In equation (2.6), we have the likelihood for the model, which is

\[ L(y|\mu, \lambda_y, \lambda_z, \theta) \propto |S|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(y - \mu 1)^T S(y - \mu 1)\right) \]  

(4.1)

In conjunction with the priors which we discussed for \( \mu, \lambda_y, \) and \( \lambda_z, \) we obtained the following full conditional distributions:

\[ \pi(\mu|y, \lambda_y, \lambda_z, \Sigma) \sim N((1^T S)^{-1} 1^T S y, (1^T S 1)^{-1}) \]

\[ \pi(\lambda_y|y, \mu, \lambda_z, \Sigma) \propto |S|^{\frac{1}{2}} \lambda_y^{a_y - 1} \exp\left(-\frac{1}{2}(y - \mu 1)^T S(y - \mu 1) - b_y \lambda_y \right) \]
\[ \pi(\lambda_z|y, \mu, \lambda_y, \Sigma) \propto |S|^{\frac{d}{2}} \lambda_z^{d^2 - 1} \exp\left(-\frac{1}{2}(y - \mu \mathbf{1})^T S(y - \mu \mathbf{1}) - b_{\lambda_z} \lambda_z\right) \]

In chapter 3, we discussed the parameters that determine the covariance structure of the model and chose appropriate prior distributions.

\[ \psi_x \sim N(0, \Sigma_{\tau_0}) \]
\[ \psi_y \sim N(0, \Sigma_{\tau_0}) \]
\[ \tau \sim N(\mu_\tau, \frac{1}{\lambda_\tau} \Sigma_{\tau_\tau}) \]

We did the same for the hyperparameters in the prior distributions for \( \psi_x, \psi_y, \) and \( \tau. \)

\[ \tau_\psi \sim \Gamma(a_{\tau_0}, \beta_{\tau_0}) \]
\[ \tau_\tau \sim \Gamma(a_{\tau_\tau}, \beta_{\tau_\tau}) \]
\[ \lambda_\tau \sim \Gamma(a_{\lambda_\tau}, \beta_{\lambda_\tau}) \]
\[ \mu_\tau \sim N(m, s^2) \]

Now, in conjunction with equation (4.1), we obtain the full conditional distributions for each of these parameters.

\[ \psi_x|y, \mu, S \propto |S|^{\frac{d}{2}} \exp\left\{-\frac{1}{2}(y - \mu \mathbf{1})^T S(y - \mu \mathbf{1})\right\} |\Sigma_{\tau_0}|^{-\frac{d}{2}} \exp\left\{-\frac{1}{2}(\psi_x^T \Sigma_{\tau_0}^{-1} \psi_x)\right\} \]
\[ \psi_y|y, \mu, S \propto |S|^{\frac{d}{2}} \exp\left\{-\frac{1}{2}(y - \mu \mathbf{1})^T S(y - \mu \mathbf{1})\right\} |\Sigma_{\tau_0}|^{-\frac{d}{2}} \exp\left\{-\frac{1}{2}(\psi_y^T \Sigma_{\tau_0}^{-1} \psi_y)\right\} \]
\[ \tau|y, \mu, S \propto |S|^{\frac{d}{2}} \exp\left\{-\frac{1}{2}(y - \mu \mathbf{1})^T S(y - \mu \mathbf{1})\right\} |\Sigma_{\tau_0}|^{-\frac{d}{2}} \exp\left\{-\frac{1}{2}(\tau^T \lambda_\tau \Sigma_{\tau_\tau}^{-1} \tau - \mu_\tau)\right\} \]

We also need the full conditional distributions for the hyperparameters, which we obtain using the prior distributions for \( \psi_x, \psi_y, \) and \( \tau \) and the prior distributions for
these parameters.

\[ \tau_y | \psi_x, \psi_y \propto \tau_x^{-1} \exp \left\{ -\beta_{\tau_y} \tau_y \right\} | \Sigma_{\tau_y} |^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left( \psi_x^T \Sigma^{-1}_{\tau_x} \psi_x + \psi_y^T \Sigma^{-1}_{\tau_y} \psi_y \right) \right\} \]

\[ \tau | \tau, \mu, \lambda \propto \tau^{-1} \exp \left\{ -\beta_{\tau} \tau \right\} | \Sigma_{\tau} |^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left( \tau - \mu \right)^T \lambda \Sigma^{-1}_{\tau} \left( \tau - \mu \right) \right\} \]

\[ \lambda | \tau, \mu, \Sigma_{\tau} \sim \Gamma \left( \frac{m}{2} + \alpha_{\lambda}, \frac{1}{2} \left( \tau - \mu \right)^T \Sigma^{-1}_{\tau} \left( \tau - \mu \right) + \beta_{\lambda} \right) \]

\[ \mu | \tau, \lambda, \Sigma_{\tau} \sim N \left( \left( \lambda^T \Sigma^{-1}_{\tau} + \frac{m}{s^2} \right) \left( 1^T \lambda \Sigma^{-1}_{\tau} + \frac{1}{s^2} \right)^{-1}, \left( 1^T \lambda \Sigma^{-1}_{\tau} + \frac{1}{s^2} \right)^{-1} \right) \]

In the full conditional for \( \lambda_{\tau} \), recall that \( m \) is the number of observation points in the data set.

### 4.2 Implementation using MCMC

Using the ideas presented in previous sections, we can implement a Monte Carlo Markov Chain procedure to sample from the full conditional distributions of all the parameters, including those that determine the matrix \( \Sigma \). The ingredients we need to do this are

- The full conditional distribution for each parameter (at least up to a constant)
- Starting values for each parameter
- Distance matrix such that the entry in row \( i \) and column \( j \) gives the distance between observation site \( i \) and observation site \( j \) (must be symmetric by definition)

Here we treat the updating scheme in a step by step fashion, discussing the difficulties encountered in updating each parameter of interest.
4.2.1 Updating $\mu$ using the Gibbs algorithm

The simplest parameter to update, $\mu$, requires only the use of a Gibbs step. Remember that the full conditional for $\mu$ is

$$
\pi(\mu|y, \lambda_y, \lambda_z, \Sigma) \sim N((1^T S1)^{-1} 1^T S y, (1^T S1)^{-1})
$$

which is of known form (multivariate normal). In this case, the Gibbs algorithm allows us to simply make random draws from the distribution, with the mean and variance determined by the current values of the other parameters. Note also that unlike many of the other parameters, updating $\mu$ does not require a concurrent update of either $\Sigma$ or $S$.

4.2.2 Updating $\lambda_y$ and $\lambda_z$ using the Metropolis-Hastings algorithm

Since the updating procedures for these two parameters are very similar, we address them together. The full conditionals in question are given by

$$
\pi(\lambda_y|y, \mu, \lambda_z, \Sigma) \propto \pi \lambda_y^{\lambda_y - 1} \exp\left\{-\frac{1}{2} (y - \mu 1)^T S (y - \mu 1) - b_{\lambda_y} \lambda_y\right\}
$$

$$
\pi(\lambda_z|y, \mu, \lambda_y, \Sigma) \propto \pi \lambda_z^{\lambda_z - 1} \exp\left\{-\frac{1}{2} (y - \mu 1)^T S (y - \mu 1) - b_{\lambda_z} \lambda_z\right\}
$$

Since these are not of known form, we need to use either the Metropolis or the Metropolis-Hastings algorithm, where this choice will depend on the desired proposal distribution. Also, note that a change in either $\lambda_y$ or $\lambda_z$ will affect the matrix $S$. Therefore, with each new proposal, denoted by $\lambda^{\text{cand}}_y$ or $\lambda^{\text{cand}}_z$, we also have to re-calculate the matrix $S$. This new matrix, $S^{\text{cand}}$, is then used to evaluate the full conditional distribution in the numerator of either the Metropolis or Metropolis-Hastings ratio. If the proposal is accepted, we not only save $\lambda^{\text{cand}}_y$ and $\lambda^{\text{cand}}_z$ as the new $\lambda_y$ or $\lambda_z$, we also have to remember to substitute $S^{\text{cand}}$ for $S$. 

40
Since $\lambda_y$ and $\lambda_z$ must be positive, it seems sensible to choose a proposal distribution that only generates positive numbers. We also decided to choose candidates dependent at least in part on the current value of the parameter. Two common possibilities are the gamma distribution and the uniform distribution (on the interval from zero to some other positive number, not necessarily one). In the case of the gamma distribution, with mean $\frac{\alpha}{\beta}$ and variance $\frac{\alpha}{\beta^2}$, one obvious way to do this might be to set the parameters $\alpha$ and $\beta$ so the mean equals the current value and the variance is set at the desired value. However, because of the right-skewed nature of the gamma distribution, the bulk of candidates will in some cases be drawn from the candidates that are less than the current value. This problem tends to occur when the current value for $\lambda_y$ or $\lambda_z$ is small. Then, many of the values generated as candidates may be very small fractions (say, on the order of $10^{-5}$). These candidates may be quite far from the current value and have a very low acceptance probability. If this occurs, the algorithm will remain “trapped” at one value for a substantial period of time, and add a very noticeable period to the burn-in period (before reaching convergence).

A reasonable alternative is to use the uniform distribution to generate candidate values for $\lambda_y$ and $\lambda_z$. We can avoid the problem mentioned above if we use a uniform distribution with boundaries 0 and $2\lambda$, where $\lambda$ is the current value of $\lambda_y$ or $\lambda_z$. In this way, we assure that there is an equal probability of generating candidate values less than or greater than the current value, and we are less likely to get trapped at one value of $\lambda_y$ or $\lambda_z$. However, if the current value becomes very low, the number of acceptances can be expected to rise, as the algorithm will be limited to small step sizes. It can take some time to move to the higher ranges, because a smaller range of values will be generated (since $2\lambda$ has decreased in accordance with $\lambda$). Also, for high values of $\lambda_y$ or $\lambda_z$, the range of candidates greatly increases, and it is possible that proportionately more candidate values will be rejected.
Given these concerns, we decided to use the uniform proposal distributions

\[
\lambda_y^{\text{and}} \sim U(0, 2\lambda_y^{\text{curr}}) \\
\lambda_z^{\text{and}} \sim U(0, 2\lambda_z^{\text{curr}})
\]

These distributions do not provide the symmetry needed to perform a Metropolis step. Since \(\pi(\lambda_y^{\text{curr}} | \lambda_y^{\text{and}}) \neq \pi(\lambda_y^{\text{and}} | \lambda_y^{\text{curr}})\) and \(\pi(\lambda_z^{\text{curr}} | \lambda_z^{\text{and}}) \neq \pi(\lambda_z^{\text{and}} | \lambda_z^{\text{curr}})\), we need to use Metropolis-Hastings steps in the updates of these two parameters. Although we have already defined the probability of drawing a particular candidate, given the current value, we need to know the probability density function for a jump in the reverse direction. That is, we need \(\pi(\lambda_y^{\text{curr}} | \lambda_y^{\text{and}})\) and \(\pi(\lambda_z^{\text{curr}} | \lambda_z^{\text{and}})\). These probabilities are

\[
\pi(\lambda_y^{\text{curr}} | \lambda_y^{\text{and}}) = \begin{cases} 
\frac{1}{2\lambda_y^{\text{and}}} & \text{if } \frac{\lambda_y^{\text{curr}}}{2} < \lambda_y^{\text{and}} < 2\lambda_y^{\text{curr}} \\
0 & \text{otherwise}
\end{cases}
\]

\[
\pi(\lambda_z^{\text{curr}} | \lambda_z^{\text{and}}) = \begin{cases} 
\frac{1}{2\lambda_z^{\text{and}}} & \text{if } \frac{\lambda_z^{\text{curr}}}{2} < \lambda_z^{\text{and}} < 2\lambda_z^{\text{curr}} \\
0 & \text{otherwise}
\end{cases}
\]

Notice that, according to the above restrictions, if the candidate \(\lambda_y^{\text{and}}\) or \(\lambda_z^{\text{and}}\) that is proposed is less than \(\frac{1}{2}\lambda_y^{\text{curr}}\) or \(\frac{1}{2}\lambda_z^{\text{curr}}\), then that candidate cannot be accepted. Since this proposal distribution is not symmetric, we must use the Metropolis-Hastings algorithm, which takes into account the proposal distribution when the acceptance ratio is calculated.

4.2.3 Updating the vectors \(\psi_x, \psi_y,\) and \(\tau\)

The more complicated part of our MCMC scheme involves updating the parameter vectors \(\psi_x, \psi_y,\) and \(\tau\). The values for these parameters affect the value of \(\Sigma\), and not just the value of \(S\), which means that there will be more complexity in the updating steps for these vectors. If the number of observation sites is large, the size of the
matrices will increase, making these calculations slower. Therefore, it is critical to choose a proposal distribution that keeps the calculation of the Metropolis-Hastings as simple as possible and that allows the chain to sample the space adequately in as few iterations as possible.

Another concern is that these parameters typically take a long time to converge. It is difficult to optimize the “step size” of the proposal distribution, so that an optimal fraction of candidates can be accepted. The appropriate proposal distribution parameters may differ from problem to problem. To address this issue, we allow for two to three update opportunities for each of the $\psi_x$, $\psi_y$, and $\tau$ vectors in each iteration of the chain, each of which is explained below.

**Large-scale update**

In each iteration, we allow each element of both $\psi$ vectors or each element of the $\tau$ vector to be updated by some randomly generated increment. We generate this increment from a uniform proposal distribution, allowing us to use the Metropolis algorithm and thus avoid the extra calculations inherent when the proposal distribution must be evaluated in the ratio of the Metropolis-Hastings ratio. This procedure allows the ellipses to rotate in the same way together, which could be important if the site has a predominant axis along which spatial dependence is structured. The whole candidate vector(s) is evaluated as a candidate in one step, so that the whole $\Sigma$ and $S$ matrices are calculated, used in the calculation of the ratio, and accepted or rejected with each iteration.

**Graduated update**

It is useful to have another update step in order to propose candidate vectors for $\psi_x$ and $\psi_y$ or $\tau$ such that the elements of the candidate vector(s) have different values.
One way to do this is to make a series of what we will call “graduated updates”, one for each member of the vector of interest. For some observation site $i$, we generate an increment from a uniform proposal distribution (as in the “large-scale” update). The $i$th member of the candidate vector is then incremented by this amount. Next, we calculate the height of an isotropic kernel centered at site $i$ at each of the other observation points, in proportion to the height at site $i$. (Note that this kernel is the same for each iteration and for each location at which it is centered. Also, note that the chosen kernel is not required to be isotropic; however, in this case this choice seems reasonable since it is easy to visualize and since the priors on $\psi_x$, $\psi_y$, and $\tau$ all include isotropic covariance structures.) For each member $j \neq i$ of the candidate vector, we increment element $j$ with a proportion of the same increment used for site $i$, based on the proportional height of the $i$th kernel at the $j$th site. This procedure generates a candidate vector such that the locations at which the parameters have changed most are sites that are relatively close together. Since the sites in which the proposal is most different from the current parameter are likely to be fairly highly correlated (due to their proximity), they are more likely to make a relatively large jump at the same time. This generally allows for more movement, although that movement is often only notable in one localized region at a each update. Then, we proceed with calculation of the Metropolis ratio as usual (for which we need to recalculate both $\Sigma$ and $S$). Note that even though this updating algorithm focuses around one site $i$, all sites are updated, though by smaller increments, when a graduated proposal is accepted. So, it is still often effective to perform a limited number of graduated updates in every iteration, rather than a graduated update focused on every element of the vector in each iteration. In our implementation, we randomly choose at the beginning of each iteration which locations will be the ones at which the kernel is centered during the graduated update steps.
Several warnings come to our attention when we use this method. Note that all the members of the vector in question must be changed in the same direction, whether positive or negative. This is inherent in the approach. So far we have not mentioned any updating procedure that would allow the members of the vector to diverge; the methods have only allowed incremental changes in either the positive or negative direction. Efficiency may be another issue. For each problem, it may be necessary to explore the best parameters to use for the proposal distribution that results in an acceptable proportion of acceptances. Since the “graduated update” involves re-making a candidate vector, candidate $\Sigma$, and candidate $S$ for each attempted update, a sizeable number of calculations must be made. It is important that we minimize that number by choosing a uniform distribution with parameters that will allow appropriate increments for proposals.

**Update one at a time**

The chief benefit of the “graduated update” strategy is that each element of the vector is given an opportunity to change more dramatically than its neighbors. However, every time the value of the vector corresponding to site $i$ changes, its neighbors also change. This update step is likely to result in more acceptances for a given stepsize because its neighbors are changing a little bit in accord. That behavior is preferred by the prior definition of some spatial dependence, as quantified in the covariance matrix $\Sigma_{\tau_0}$. However, as mentioned above, we have to allow an opportunity for the elements of the vector to vary in a different direction from the others, even if we have to use a smaller stepsize in the proposal distribution.

For each observation site $i$, we generate a proposal for the $i$th vector element. In order to simplify the calculations we generate a candidate from the marginal prior density of the $i$th element. This means that the prior densities will “cancel out” in
the Metropolis-Hastings ratio, and some time will be saved. In the candidate vector, only the element corresponding to the \( i \)th site is changed. However, since we will have to use this procedure for to update the vector at each site, it is important to minimize as much as possible the calculations that we need to calculate the ratio.

The candidate vector(s) being proposed is only different from the current one(s) in a combination \( \psi_x \) and \( \psi_y \) (for a site \( i \)) or one \( \tau \) (for a site \( i \)). Since only one kernel is being modified, only one line and one column of the matrix \( \Sigma^{\text{cand}} \) is different from the the current \( \Sigma \). In accordance, the only rows and columns of \( S \) that need to be updated in the \( S^{\text{cand}} \) are those that correspond to observations made at site \( i \). Unless there are a large number of observations taken at a site, this should be a very small portion of the \( S \) matrix.

In order to calculate the Metropolis-Hastings ratio, however, we also need to calculate the determinant of \( S \). This is normally accomplished by taking the Cholesky decomposition of \( S \), and then multiplying the diagonal elements in order to achieve the desired quantity \( |S|^{1/2} \). Further, this Cholesky decomposition is used to calculate the quantity in the exponent, while avoiding direct calculation of the inverse of \( S \). The standard routine for calculating the Cholesky decomposition, \( L \), of a matrix proceeds column by column, with the calculations for each element \( L_{ij} \) depending on the diagonal elements \( L_{ii} \) and the \( i - 1 \) previous elements in the same row \( j \). So, if the \( i \)th row and column of \( S \) changes, the Cholesky decomposition only has to be recalculated for the portion in and after the \( i \)th column. Although this is not very helpful if the first few columns have changes (this will involve changing the huge majority of elements in the matrix), it is extremely helpful if only one row and column changed in a later portion of the matrix. Calculating only the portions of the candidate matrices that we absolutely must substantially lowers the number of calculations we have to do evaluate the full conditional distribution in the numerator.
of the Metropolis-Hastings ratio.

Since each successful acceptance only changes the vector at one element, it is important that we make this step as efficient as possible and minimize the cost for the relatively small “payoff”. It is also possible to experiment with the individual data set and model, because in many cases, it may not be necessary to include this step each time. Executing this step on a periodic basis, perhaps every five to ten iterations, may compensate for the reality that the vast majority of movement in the chains for these vectors will come in the previous two update methods. Because this step requires that we loop through the Metropolis-Hastings calculations for each of \( m \) observation sites, minimizing this process could make the algorithm much more efficient. A good plan might be to view this step as a utility merely for “tweaking” the individual kernels more finely.

4.2.4 Updating \( \tau_{\psi} \) and \( \tau_{\tau} \)

Now we consider the parameter \( \tau_{\psi} \), which governs the spatial dependence between the eccentricity and orientation of the ellipses, and \( \tau_{\tau} \), which governs the spatial dependence for the effective size of the ellipses. These two parameters can be addressed together, since their roles are very similar. We can calculate the the full conditional distributions as

\[
\tau_{\psi} | \psi_x, \psi_y \propto \tau_{\psi}^{-\alpha_{\tau_{\psi}}-1} \exp\left\{ -\beta_{\tau_{\psi}} \tau_{\psi} \right\} | \Sigma_{\tau_{\psi}} |^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \left( \psi_x^T \Sigma_{\tau_{\psi}}^{-1} \psi_x + \psi_y^T \Sigma_{\tau_{\psi}}^{-1} \psi_y \right) \right\}
\]

\[
\tau_{\tau} | \tau, \mu_{\tau}, \lambda_{\tau} \propto \tau_{\tau}^{-\alpha_{\tau_{\tau}}-1} \exp\left\{ -\beta_{\tau_{\tau}} \tau_{\tau} \right\} | \Sigma_{\tau_{\tau}} |^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (\tau - \mu_{\tau})^T \lambda_{\tau} \Sigma_{\tau_{\tau}}^{-1} (\tau - \mu_{\tau}) \right\}
\]

which are both of unknown form. We use a normal distribution in order to generate proposals in each case. Since this has a symmetric probability distribution function, we need to only compute the Metropolis ratio to help us decide whether the proposal will be accepted. This involves the calculation of \( \Sigma_{\tau_{\psi}} \) and \( \Sigma_{\tau_{\tau}} \) and their Cholesky
decompositions given the candidate values. If the proposal values are accepted, we also adjust these matrices accordingly, maintaining these matrices from iteration to iteration with the current values of $\tau_\psi$ and $\tau_\tau$.

### 4.2.5 Updating $\mu_\tau$ and $\lambda_\tau$

As discussed previously, $\lambda_\tau$ is the parameter that represents the marginal precision for each element of the vector $\tau$. Since $\tau$ is distributed according to the multivariate normal distribution and since $\lambda_\tau$ has a gamma prior (the conjugate prior for the normal precision parameter), $\lambda_\tau$ is of known form. At each iteration, we can use a Gibbs step to generate a new value for $\lambda_\tau$ using the following gamma distribution.

$$
\lambda_\tau|\tau, \mu_\tau, \Sigma_{\tau_\tau} \sim \Gamma(m + \alpha_{\lambda_\tau}, \frac{1}{2}(\tau - \mu_\tau \mathbf{1})^T \Sigma_{\tau_\tau}^{-1}(\tau - \mu_\tau \mathbf{1}) + \beta_{\lambda_\tau})
$$

With the change in $\lambda_\tau$, no concurrent updates of matrices (such as $\Sigma$ or $\Sigma_{\tau_\tau}$ are required.

The parameter $\mu_\tau$, which represents the mean of the prior distribution on $\tau$ (recall that all elements of the mean vector are assumed to be the same), is similarly easy to update. Since we have a normal prior on $\mu_\tau$, this leads us to the conjugate full conditional on $\mu_\tau$, which is also normal.

$$
\mu_\tau|\tau, \lambda_\tau, \Sigma_{\tau_\tau} \sim N((\lambda_\tau \tau^T \Sigma_{\tau_\tau}^{-1} + \frac{m}{s^2})(1^T \lambda_\tau \Sigma_{\tau_\tau}^{-1} \mathbf{1} + \frac{1}{s^2})^{-1}, (1^T \lambda_\tau \Sigma_{\tau_\tau}^{-1} \mathbf{1} + \frac{1}{s^2})^{-1})
$$

Once again, we use a Gibbs step based on this distribution to obtain a new value for $\mu_\tau$ at every iteration. Note that, like $\lambda_\tau$, our update of $\mu_\tau$ does not require us to update other matrices or vectors.

Although we have formulated the full conditionals for $\tau_\tau$ and $\lambda_\tau$ in the past two sections, it should be noted that we have not yet been consistently successful in including these two parameters in our MCMC updating scheme. It seems that each
of these parameters needs a fairly informative prior in order to serve its function in the model. Use of priors that are not very informative often leads to “over-fitting” of the data and related issues. This is one obvious area for extensions to the current work and is discussed more fully in Chapter 7.

4.3 Estimating the spatial process

In the preceding section, we discussed the implementation of MCMC ideas to sample from the full conditional distribution of each of the parameters specific to the determination of the correlation matrix $\Sigma$. However, in many applications the subject of interest may be the values taken by the Gaussian process at various sites within the region. At sites for which we have observations, this information is denoted by the vector $z$. Since earlier we integrated $z$ out of the model, the updates of the other parameters have not depended on it. It was not necessary to update the $z$ vector as we went along. Now, we will need to incorporate some sampling from the full conditional for $z$ in order to gain some understanding of the spatial process in the region of interest. There are some sites for which we do not have observations, and it may be desirable to estimate the value of the spatial process, which we call $z'$, at these points as well. This is more difficult computationally, and we address these concerns in the following sections.

4.3.1 Sampling from the full conditional distribution of $z$

When we discussed the model in a previous chapter, we noted in equation (2.5) that based on a combination of the prior distribution on $z$ and the likelihood of the model, the full conditional distribution for the $z$ vector can be written as

$$
(z|\Sigma, \lambda_y, \mu, y, \lambda_z) \sim N(\lambda_y(\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1} D^T (y - \mu 1), (\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1})
$$
Recall that $D$ is a matrix that serves to keep track of which observations, if any, were taken at the same site. If each observation comes from a different site, as is the case in many examples, this matrix is simply the identity matrix.

Using equation (2.5), we can generate realizations from the full conditional distribution for $z$. Since $z$ is not used in the updating procedures for the other parameters, we do not need to update it at every iteration. For instance, it is not necessary to sample from the full conditional for $z$ until the other parameters have converged to their posterior distributions, since these realizations would not likely be from the true posterior distribution of $z$. Also, if the number of iterations is large, we only need to perform this step periodically, obtaining only as many sample values from the posterior distribution as we feel are necessary.

4.3.2 Difficulties encountered when using the full conditional distribution for $z$

One complication that can arise when we attempt to calculate the parameters for the distribution given in (2.5) is the inability to obtain $\Sigma^{-1}$. This occurs when the spatial process realizations for two different observation sites $i$ and $j$ are strongly correlated; this is mostly likely to happen when the points $i$ and $j$ are a very short distance from each other. As a result, the rows $i$ and $j$, and by extension the columns $i$ and $j$ (remember that $\Sigma$ is symmetric), are very similar. In such cases, it is possible that a computer will not be able to complete the matrix inversion, or if the inversion is completed, a significant amount of numerical error is involved. This is due to the fact that the pivots for $\Sigma$ are very small, and the computer does not have enough precision to record a non-zero (but extremely small) pivot or to record enough of the significant digits to keep the error within reasonable bounds.

We propose a method to deal with this problem, and develop it more fully in
Chapter 5. This involves using a special version of the Cholesky algorithm that finds
the number of pivots with value greater than some given tolerance \( TOL \). We then
choose the rows (and columns) corresponding to these “good” pivots, and also the
 corresponding observed values. We generate from the full conditional for \( z \) given in
(2.5) only for these chosen sites, and then use these values to determine the elements
of \( z \) corresponding to the other sites as a linear combination of the \( z \) elements at
the good sites. The “weights” for this calculation are given by our special Cholesky
algorithm for the matrix \( \Sigma \). This difficulty and our solution are explained in more
detail in a chapter 5.

4.3.3 Sampling from the distribution of \( z' \)

In the previous section, we learned how to sample from the posterior distribution of
the spatial process at sites where we had observed data. However, in many appli-
cations, it is desirable to estimate the value of the spatial process at sites within
the region at which no data was observed. For instance, we may want to get an
idea of what the spatial process is like over the whole region of interest. One way
to do this would be to establish a grid of closely spaced sites over the region, and
sample from the posterior distribution of the spatial process at all these points. Then
this data can be easily used in conjunction with many different software packages to
generate contour plots or other graphics that neatly summarize this spatial process
to an audience.

Joint distribution for \( z \) and \( z' \)

In order to do this, we need to establish the distribution for \( z' \). It is useful to think
of the joint distribution of \( z \) and \( z' \). We can extend our prior for \( z \), since there is no
reason to expect that \( z' \) will behave in a different pattern. Then, we write

\[
\begin{pmatrix}
  z' \\
  z''
\end{pmatrix} \sim N\left(0, \frac{1}{\lambda_z} \Sigma_{total}\right) \tag{4.2}
\]

where \( \Sigma_{total} \) can be written more clearly as

\[
\Sigma_{total} = \begin{pmatrix}
  \Sigma & \Sigma' \\
  \Sigma' & \Sigma''
\end{pmatrix}
\tag{4.3}
\]

Notice that \( \Sigma_{total} \) can be formed using three pieces. We already know the first, \( \Sigma \), which is the correlation matrix between realizations of the spatial process at the observed sites. Another is \( \Sigma' \) whose elements consist of correlations between unobserved site \( i \) and observed site \( j \). The transpose of this matrix is also used in the upper right portion of \( \Sigma_{total} \). The last piece is \( \Sigma'' \), with entries given by the correlation of the spatial process among the unobserved sites \( i \) and \( j \).

**Conditional distribution of \( z'|z \)**

Since the joint distribution given by (4.2) is multivariate normal, it is straight-forward to find the conditional distribution of one part given the other. We know that \( z'|z \) must also be normal with certain parameters as shown below

\[
z'|z \sim N\left(\Sigma'\Sigma^{-1}z, \Sigma'' - \Sigma'\Sigma^{-1}\Sigma'\right) \tag{4.4}
\]

Now, we can generate from this distribution after each random generation we make from the full conditional for \( z \). Another option is to generate from the joint distribution given by (4.2). In this case, we would have to allocate computer memory for a matrix composed of four parts: \( \Sigma \), \( \Sigma'^T \), \( \Sigma' \), and \( \Sigma'' \). Most computers generate from the multivariate normal distribution using the Cholesky decomposition of the covariance matrix, which is a time-consuming procedure for a matrix as large as the joint covariance matrix \( \Sigma_{total} \) is. The procedure we use here also requires taking the
Cholesky decomposition of the matrix $\Sigma$, but this matrix is much smaller. It is also simpler for us to calculate the covariance matrix in its requisite pieces, and we use it in this same form (without the need of copying this information into a larger matrix).

**Calculating $\Sigma'$ and $\Sigma''$**

In previous sections, we realized that calculating $\Sigma_{\text{total}}$ was crucial to our calculation of $z'$. However, at this point in the updating scheme, we only have $\Sigma$ available. We must obtain $\Sigma'$ and $\Sigma''$ to continue.

Remember that our calculation of $\Sigma$, since it is based on the process convolution approach (as described in Chapter 3), requires the input of the coordinates of a focus point $(x, y)$ and a scaling factor $c$. Up until this point, we have only obtained these parameters for locations at which observations have been made. We need to obtain the corresponding parameters for sites at which no data has been taken.

Recall that the priors for the parameters $\psi_x$, $\psi_y$, and $r$ can be written as

$$\psi_x \sim N(0, \Sigma_{\psi})$$

$$\psi_y \sim N(0, \Sigma_{\psi})$$

$$r \sim N(\mu_r, \frac{1}{\lambda_r} \Sigma_{\psi})$$

Since each of these distributions is multivariate normal (as was the case with the joint distribution of $z$ and $z'$), we can obtain a realization for these sets of parameters using the same method we discussed in conjunction with equation (4.4). Since both $\Sigma_{\psi}$ and $\Sigma_r$ both correspond to Gaussian covariance structures, we only need the distances between all pairs of observed points and prediction points and the distances between the prediction points to calculate the parameters for the multivariate normal conditional distributions $\psi_x | \psi_y, \psi'_y | \psi_y$, and $r | r'$.

Once we have a realization of $\psi'_x$, $\psi'_y$, and $\Sigma'$, we can use the parameters to calculate
\( \Sigma' \) and \( \Sigma'' \) as if we were calculating \( \Sigma \). Then these submatrices can be substituted into equation (4.4), and we can complete our original goal, which was obtaining a realization from the spatial process at locations for which no data is available.
Chapter 5

Computing with semi-positive definite covariance matrices

5.1 Background concerning positive matrices

Recall that a symmetric matrix $A$ is said to be positive definite matrix if and only if for any vector non-zero vector $x$

$$x^T A x > 0$$

(5.1)

From this statement, we can derive that the following are also necessary and sufficient conditions for us to determine that $A$ is positive definite (Strang, 1988).

1. All the pivots of $A$ are positive.

2. All the eigenvalues of $A$ are positive.

Note that all multivariate normal covariance matrices (and their inverses, which are precision matrices) must be positive definite.

For any symmetric positive definite matrix $A$, we can find a lower triangular matrix $L$ such that $A = LL^T$. This is called the Cholesky decomposition of the matrix $A$. Finding the Cholesky decomposition of a matrix is critical for us to perform needed computations, such as finding the determinant of $A$ and randomly
generating from a multivariate normal distribution with covariance matrix $A$. More details about the Cholesky decomposition, and the algorithm to calculate it can be found in the next section.

A semi-positive definite matrix has a similar definition, with the exception that we relax the inequality to allow a zero result.

$$x^T Ax \geq 0 \quad (5.2)$$

Here, we can encounter pivots that are equal to zero, as well as eigenvalues that are equal to zero. Even if a matrix is in fact positive definite, it can appear to the computer as semi-positive definite if the pivots are very small. This is because the computer can determine each number only to a limited level of precision. If the pivot is very small (i.e., it is different from zero in the digits that are beyond a computer’s level of precision), calculation of the pivot may seem to produce a pivot with a value of zero. Even if the computer recognizes that the pivot is not actually zero, the closer the pivot’s value is to zero, the more round-off error can be expected in certain subsequent computations (such as finding the inverse) as a result.

### 5.2 How the Cholesky algorithm works

We now review how the Cholesky algorithm works on a correlation matrix like $\Sigma$ to produce the lower triangular matrix $L$. We proceed through the matrix column by column, performing the requisite operations using the elements calculated before. Each element of the matrix $L_{ij}$ is calculated using the pivot for its column $L_{jj}$ and the previous columns’ entries in the $i$th and $j$th rows.

The basis of the algorithm is as follows:

1. Set all the elements in the upper triangle equal to 0.

   $L$ will be a lower triangular matrix.
2. Let \( L_{11} = \sqrt{\Sigma_{11}} = 1. \)
   
   Our starting point is the correlation between \( z_1 \) and \( z_1 \) which must be 1.

3. For \( i = 2, \ldots, j \), set \( L_{i1} = \frac{\Sigma_{i1}}{L_{11}} = \Sigma_{i1} \)
   
   These are the correlations of the other \( z_i \)s with \( z_1 \).

4. Proceed through the other columns left to right, one at a time
   
   (a) For all other diagonal elements \( L_{ii} = (\Sigma_{ii} - \sum_{k=1}^{i-1} L_{ki}^2)^{1/2} \). We find the diagonal elements, we are actually calculating the “amount” of correlation left, given that the original total was 1. This is done by subtracting from the diagonal element the squares of all the correlations of \( z_i \) with the \( i - 1 \) sites before, and then taking the square root. If row \( b \) is highly correlated with \( a \), then this diagonal element is very small, and, if \( z_a \) and \( z_b \) are perfectly correlated (or seem so, due to lack of computer precision), the pivot is zero.

   (b) For all other non-diagonal elements \( L_{ji} = \frac{1}{L_{ii}} \sum_{k=1}^{i-1} L_{ik}L_{jk} \). In this step, if the pivot \( L_{ii} \) previously calculated was zero, the algorithm fails, since division by zero is undefined. If the pivot was close to zero, with some of the digits beyond the computer’s storage capacity, round-off error will result. The closer the pivot is to zero, the more significant digits will be lost, and the more extreme the round-off error will be.
5.3 A potential problem when sampling from the full conditional for $z$

Recall that our model gives the prior distribution for the spatial component of the process as

$$z \sim N(0, \frac{1}{\lambda_z} \Sigma)$$

In this case $\Sigma$ is the correlation matrix. It must be a positive definite matrix, and it is determined by our process convolution method described earlier. Let us suppose that the matrix is of dimension $n$ by $n$, and that among the $n$ sites are three denoted by $a$, $b$, and $c$. The rows and columns corresponding to covariance of the spatial process at these points are rows (and columns) $a$, $b$, and $c$. Let us suppose that the measurements at $z_a$ and at $z_b$ are highly correlated; this could happen, for instance, if the sites $a$ and $b$ are very close to one another. Then, the rows and columns corresponding to $a$ and $b$ will be nearly the same. The more similar the rows are, the closer the pivots are to zero. As discussed above, this can result in $\Sigma$ being effectively semi-positive definite for the purposes of further computing, such as taking the inverse of $\Sigma$.

The inverse of $\Sigma$ is needed to generate random samples from the full conditional distribution of the $z$ process at observed sites as well as at unobserved sites, based on the $n$ sites at which observations were taken. This procedure is discussed in more detail in chapter 3. In order to avoid problems with finding $\Sigma^{-1}$, we develop a version of the Cholesky algorithm to help us determine if the matrix’s pivots are small enough to present a problem, and if so, we present an approach to the problem.
5.4 A modified Cholesky algorithm for use with semipositive definite matrices

The two common approaches to the problem described above are the pivoting approach and the tolerance based approach. The pivoting approach differs from the ordinary algorithm in that we determine the order of the columns based on the pivots, while in the traditional approach described above, we address each column in order. This minimizes the roundoff error by choosing the pivots in order of magnitude. However, if some of the pivots are very small, an unacceptable amount of roundoff error may still result. On the other hand, the tolerance-based approach treats the columns in the standard order, but if a pivot is calculated with a tolerance lower than a given threshold, the operation completely stops. In this case, it is possible that if the pivoting order was suitably chosen, the algorithm could have progressed further. Both of these algorithms have already been implemented in FORTRAN in the IMSL libraries.

In our application, an optimal approach would be to combine the good features of each of the previous algorithms. We want to determine the order of operations based on the pivots, taking the largest pivot each time, and terminate the algorithm when no pivots larger than a specified tolerance $TOL$ remain. By dealing with the pivots in order of magnitude, we can minimize the effect of round-off error, as the smaller pivots will deal with last and therefore used in the smallest possible number of calculations. When we reach a pivot with a value less than the specified tolerance, we will know that no larger pivots are remaining, and terminate the procedure. At that point, we will know how many pivots exist that meet the required tolerance specification and proceed accordingly.

1. Initialize $L$ by setting all elements equal to 0.
2. Initialize arrays. For \( i = 1, \ldots, n \), \( d_i = \sum_{j=1}^{i} \), \( u_i = -1 \), \( p_i = -1 \), \( r = 0 \).

3. Initialize \( r = 0 \). As we find pivots we change \( r \), which gives the rank of \( L \).

4. For \( i = 1, \ldots, n \), do the following:

   (a) Find the maximum value in \( d \), denoted by \( d_j \), such that \( u_j = -1 \).

      * If \( d_j < TOL \), then EXIT. All potential pivots are too close to 0.

      * Else \( d_j \geq TOL \), then CONTINUE.

   (b) Increment \( r \) by 1.

   (c) Set \( p_i = j \) and \( u_j = i \).

   (d) Set \( L_{jp_i} = d_i \). This is the new pivot.

   (e) For \( m = i, \ldots, n \) and \( m \neq p_i \), \( L_{mp_i} = \frac{\sum_{k=1}^{m-1} L_{mp_k} L_{p_k p_i}}{L_{jp_i}} \).

   (f) For \( m = 1, \ldots, n \) and \( u_m = -1 \), set \( d_m = (d_m^2 - (L_{mp_i})^2)^{1/2} \).

5. Return the completed \( L \) matrix, the array \( p \), and the rank \( r \) of the matrix (given the tolerance \( TOL \)).

The first step of the algorithm performs a series of initializations. We will need more variables to handle the increased amount of bookkeeping necessary to determine how to choose the pivots and to perform the remaining calculations accordingly. The array \( d \), of size \( n \), is used to keep track of the value of the diagonal elements of the Cholesky decomposition as we go along. We initialize \( d \) with the square roots of the diagonal elements of the \( \Sigma \) matrix. Then, we create a new array \( u \) to keep track of which columns have already been designated as pivoting columns and initialize this column with the value \(-1\). Using this value assures us that it will be difficult to confuse the initialization value with a legitimate one. The array \( p \) keeps track of the order in which columns are designated as pivots; this information is used in
calculations later in the algorithm. For instance, if column 3 contains the second pivot, \( p_2 = 3 \) and \( u_3 = 2 \). The vector \( p \) will be returned to the calling program when this subroutine is completed. This is because it may be needed to determine which columns had a diagonal element that served as a pivot and to determine the order in which the columns were chosen.

The notable part of the algorithm is found in step 4a above. We look at each element of the array \( d \) to determine which column contains the largest diagonal element that can be used as a pivot. There are only two main restrictions on this choice. The first is that we cannot choose a column from which the diagonal element has already been used as a pivot. Also, we must be sure that the potential pivot is larger than the given tolerance \( TOL \). If there are no pivots that meet this condition, the algorithm terminates, since the user has determined that the use of any pivot smaller than \( TOL \) will lead to an unacceptable amount of computational error.

After the pivot is chosen, two more sets of computations must be performed. All the entries in the pivoting column which are in rows corresponding to the columns that have not yet served as pivots are calculated. So, if we have just calculated the \( i \)th pivot, we will have to calculate \( n - i \) other entries in the column. These entries depend on the pivot for the column as well as the other entries that are on the same row or on the row corresponding to the pivot. The last step is in preparation to find the next pivot; if \( n \) pivots have already been found, this step is not necessary, and the subroutine is terminated. For each element, in the array of diagonal elements \( d \), we subtract from its squared value the squared value of the entry in \( L \) that is found in its row and the column of the latest pivot. Then, we take the square root of these values, and begin the process again at step 4, with the comparison of the entries in the array \( d \).
5.5 Comparison between the new and old approaches with an example

Having discussed both approaches, we apply each approach to a simple example. We contrast the two approaches in this instance.

Consider a one-dimensional example of four observations taken along a line. If we trace this line our “x-axis”, the observations would be made at points 2, 2.5, 2.01, 3, 10. We can use the distances between each set of points $i$ and $j$ to fill in a distance matrix, where the element in row $i$ and column $j$ is the distance between point $i$ and point $j$. This gives

$$
\begin{pmatrix}
0.00 & 0.50 & 0.01 & 1.00 & 8.00 \\
0.50 & 0.00 & 0.49 & 0.50 & 7.50 \\
0.01 & 0.49 & 0.00 & 0.99 & 7.99 \\
1.00 & 0.50 & 0.99 & 0.00 & 7.00 \\
8.00 & 7.50 & 7.99 & 7.00 & 0.00
\end{pmatrix}
$$

If we are using a Gaussian correlation function of the form

$$
Corr(z_i, z_j) = \exp\left\{ -\frac{d_{ij}^2}{\tau^2} \right\}
$$

with $\tau = 20$, then the correlation matrix is

$$
\begin{pmatrix}
1.0000000 & 0.993752 & 0.9999998 & 0.9975031 & 0.8521438 \\
0.993752 & 1.0000000 & 0.9993999 & 0.9993752 & 0.8688151 \\
0.9999998 & 0.9993999 & 1.0000000 & 0.9975527 & 0.8524845 \\
0.9975031 & 0.9993752 & 0.9975527 & 1.0000000 & 0.8847059 \\
0.8521438 & 0.8688151 & 0.8524845 & 0.8847059 & 1.0000000
\end{pmatrix}
$$

Now, we demonstrate both the traditional Cholesky algorithm and the modified Cholesky algorithm on this sample correlation matrix, and compare the results.

Recall that when using the traditional algorithm, we proceed column by column, from left to right. Since the diagonal element for the first column is one, the first
column of the Cholesky matrix will be the same as the first column of the correlation matrix. In the modified algorithm, we compare the square roots of all the diagonal elements to find the largest one. Since all the diagonal elements are one in this case, the first column is chosen by default. This leaves the first column of the correlation matrix to also be the first column of the modified Cholesky matrix.

In the second step, we see some differences between the two procedures. The traditional algorithm proceeds directly with the second column, obtaining

\[
\begin{pmatrix}
  0.000000 \\
  0.0353442934 \\
  0.0007068938 \\
  0.0706002950 \\
  0.4867459246 \\
\end{pmatrix}
\]

However, the modified algorithm calculates what the second pivot would be in the case that the diagonal element from each of the remaining columns (the first column has already provided one pivot) is used. This list is

0.0353443 0.0007071 0.0706224 0.5233077

in order of the columns from which each diagonal element is derived (left to right). The biggest of these, 0.5233077, is then chosen, and therefore the last column is addressed next. The portions of the decompositions that have been obtained up to this point are displayed in Figure 5.1. Note that the dots are place-holders for the portions of the decompositions which have not yet been calculated.

In the next iteration of the traditional algorithm, we can see the problem beginning to form. We calculate the pivot of the third column, and find that it is already quite low, at 0.0000173. We are drawing close to our tolerance value, with still two pivots left to find. However, in the modified algorithm our choices for the third pivot are

0.0129792 0.0002749 0.0243561
\[
\begin{pmatrix}
1.000000 & 0.000000 & \cdots \\
0.9993752 & 0.0353443 & \cdots \\
0.9999998 & 0.0007069 & \cdots \\
0.9975031 & 0.0706003 & \cdots \\
0.8521438 & 0.4867459 & \cdots \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
1.000000 & \cdots & 0.000000 \\
0.9993752 & \cdots & 0.0328749 \\
0.9999998 & \cdots & 0.0006515 \\
0.9975031 & \cdots & 0.0662895 \\
0.8521438 & \cdots & 0.5233077 \\
\end{pmatrix}
\]

**Figure 5.1:** After two iterations: (top) Results from the traditional algorithm; (bottom) Results from the modified algorithm.

So, we are able to chose 0.0243561, obtained from the diagonal element in the fourth column, as our next pivot. This pivot is a power of 10^2 larger than that used by the traditional algorithm. Now we have completed three columns of the decomposition for each of the two matrices, as shown in Figure 5.2.

\[
\begin{pmatrix}
1.000000 & 0.000000 & 0.000000 & \cdots \\
0.9993752 & 0.0353443 & 0.000000 & \cdots \\
0.9999998 & 0.0007069 & 0.000173 & \cdots \\
0.9975031 & 0.0706003 & -0.0017647 & \cdots \\
0.8521438 & 0.4867459 & -0.0181515 & \cdots \\
\end{pmatrix}
\]

**Figure 5.2:** After three iterations: (top) Results from the traditional algorithm; (bottom) Results from the modified algorithm.

Things progress normally using each method, until we try to find the last pivot.
\[
\begin{pmatrix}
1.000000 & 0.000000 & 0.000000 & 0.000000 \\
0.9993752 & 0.0353443 & 0.000000 & 0.000000 \\
0.9999998 & 0.0007069 & 0.000173 & 0.000000 \\
0.9975031 & 0.0706003 & -0.0017647 & 0.0000714 \\
0.8521438 & 0.4867459 & -0.0181515 & 0.0714183 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
1.000000 & 0.000000 & 0.000000 & 0.000000 \\
0.9993752 & 0.0002674 & 0.0129765 & 0.0328749 \\
0.9999998 & 0.0000114 & 0.0002746 & 0.0066515 \\
0.9975031 & 0.0000000 & 0.0243561 & 0.0662895 \\
0.8521438 & 0.0000000 & 0.0000000 & 0.5233077 \\
\end{pmatrix}
\]

**Figure 5.3:** After four iterations: (top) Results from the traditional algorithm; (bottom) Results from the modified algorithm

Then we see the effects of numerical error. After the first four iterations, we have the matrices given in Figure 5.3.

Using the traditional approach, we square all the numbers in the last row of the decomposition that we have obtained so far (these are the first four elements in the last row), and we add the values together. Then we subtract this number from one, since one is the value in the last diagonal position, and take the square root. In this case, the final sum is:

\[
0.8521438^2 + 0.4867459^2 + (-0.181515^2) + 0.0714183^2 \approx -0.0011189
\]

Since this number is negative, the algorithm fails, and we are left without the complete Cholesky decomposition.

The alternative approach also fails to find a fifth pivot, but is able to provide better information that will allow us to work around the problem, using the solutions described in the remainder of this chapter. We follow the same basic procedure here, but we alter it a bit to allow for the fact that our pivots have not been chosen in a predetermined order. When calculating a pivot, the values that we are interested
\[
\begin{pmatrix}
1.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \\
0.9993752 & 0.0002674 & 0.0000000 & 0.0129765 & 0.0328749 \\
0.9999998 & 0.0000114 & 0.0000000 & 0.0002746 & 0.0006515 \\
0.9975031 & 0.0000000 & 0.0000000 & 0.0243561 & 0.0662895 \\
0.8521438 & 0.0000000 & 0.0000000 & 0.0000000 & 0.5233077
\end{pmatrix}
\]

**Figure 5.4:** The completed decomposition returned by the modified algorithm

in are those that are in the same row as the potential pivot. In this case, only the
diagonal element of the third column is left to become the last pivot. So, we square
the other values that are in the third row, those that are in the columns that have
already served as pivots, and add them together.

\[
0.9999998^2 + 0.0000114^2 + 0.0002746^2 + 0.0006515^2 \approx -0.0000001
\]

Once again, this is a negative number, so the algorithm is completed. However, note
that this number is over three factors of ten closer to zero (and the positive numbers),
than its counterpart in the traditional Cholesky algorithm. As shown in Figure 5.4,
its return values include the matrix values calculated so far, with all zeroes in the
third column (in which no pivot could be found). It also includes a vector that gives
the order in which the pivots were selected and a integer that gives the rank of the
decomposition matrix. In this case, those values are

\[
\text{pivot vector: } 1 \quad 5 \quad 4 \quad 2 \\
\text{rank: } 4
\]

Note that the modified algorithm will also terminate if all possible remaining pivot
values are less than the given tolerance \( TOL \). As we can see, numerical error can give
a value that is too low than the actual value of the pivot. If this happens during the
traditional algorithm, a pivoting value that is smaller than the actual analytically
calculated pivot results, but the algorithm continues to run. For a very small pivot,
this can result in a “floating exception” or “divide by zero” error, as the computer is unable to divide by a very small number. For this reason, the careful designation of a tolerance level \( TOL \) is essential to avoid numerical problems that result in the unexpected termination of a program implementing MCMC techniques.

5.6 Using the modified Cholesky algorithm in practice

We use this algorithm before each attempt to generate from the full conditional distribution for \( z \). As input we have the correlation matrix \( \Sigma \) of dimension \( n \). It is necessary to choose an appropriate tolerance \( TOL \) for the pivots, which can be based on the user’s knowledge of the computer’s precision or on the level of precision that is needed for a specific problem. Since \( z \) is not used in the updates of the other parameters (recall that it was integrated out of the likelihood function in chapter 1), it may be possible to use a tolerance that is fairly large, say \( 10^{-4} \). In addition to the Cholesky decomposition of the \( \Sigma \), the outputs from the routine are \( r \), which reveals how many pivots meet the criteria, and \( p \), which states which columns contained these \( r \) pivots. If \( r = n \), then we progress normally. Otherwise, we pursue a slightly modified updating system for the vector \( z \), as well as for \( z' \).

Now, we address the case in which \( r < n \). We use the vector \( p \) to find out which rows/columns contain qualifying pivots; the \( r \) \( z \) elements corresponding to these rows/columns can serve as a “basis” for the \( n - r \) other \( z \) elements. Eventually, we show how we are able to compute the non-basis \( z \) elements as a combination of the basis \( z \) elements. So, we separate the vector \( z \) into parts \( z_1 \), where \( z_1 \) consists of entries in \( z \) that correspond to the rows/columns that contain pivots, and \( z_2 \), which contains all other entries of \( z \). Other vectors that are used to calculate the parameters in the full conditional distribution for \( z \), whether directly or indirectly through these
parameters, must be treated similarly. These include $y$, $\psi_x$, $\psi_y$, and $\tau$; the notation for these new subvectors follows the example of $z_1$ and $z_2$. Similarly, we split $\Sigma$ into submatrices. We assign to the $r \times r$ matrix $\Sigma_{11}$ all the group 1 rows and columns, i.e. the rows and columns that contained a pivot greater than $TOL$. The Cholesky decomposition of $\Sigma_{11}$ that we have just calculated is also subsetted in the same way, producing the matrix $L_{11}$. The rows of group 2 and the columns of group 1 are selected to fill the $n - r \times r$ matrix $\Sigma_{21}$.

Having determined which columns of the matrix can serve as “basis”, we consider how to use the “basis” elements $z_1$ to determine $z_2$. The next section reviews this procedure.

5.7 Generating from the full conditional distribution for $z$

Given the partitioning of the $z$ vector discussed in the last section, we can think of our prior distribution for $z$ in a different manner. Restating equation (2.5), previously we had

$$z|\Sigma, \lambda_y, \mu, y, \lambda_z \sim N(\lambda_y (\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1}D^T(y - \mu 1), (\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1})$$

However, we recognize that this assumes that $\Sigma$ is of full rank (due to the inversion), but in our case, roundoff error makes this difficult in practice. We are working on the assumption that, for computing purposes, the matrix $\Sigma$ is actually semi-positive definite. We have divided $z$ into a “basis” portion and a “non-basis” portion, based on our results from the Cholesky decomposition. Since the $\Sigma_{11}$, which is the partition of $\Sigma$ that consists of rows and columns corresponding to the “basis” $z$ elements, is invertible, we propose a strategy for updating $z_1$ based on equation (2.5) and then updating $z_2$ based on $z_1$. We need to establish what the conditional distribution of $z_2$ given $z_1$ is for this case.
5.7.1 Conditional distribution of $z_2$ given $z_1$

We begin with a well-known result concerning the multivariate normal distribution. Suppose the vector $x$ is distributed normally with mean $\mathbf{m}$ and covariance matrix $V$. Similarly to our case, the vector $x$ is partitioned into two segments $x_1$ and $x_2$. Then we can write the distribution of $x$ as follows

$$
\begin{pmatrix}
  x_1 \\
  x_2
\end{pmatrix}
\sim
\mathcal{N}
\left(
\begin{pmatrix}
  m_1 \\
  m_2
\end{pmatrix},
\begin{pmatrix}
  V_{11} & V_{12} \\
  V_{21} & V_{22}
\end{pmatrix}
\right)
$$

Using these partitions, we can then say that the conditional distribution of $x_2$ given $x_1$ is

$$
x_2|x_1 \sim \mathcal{N}(m_2 + V_{21}V_{11}^{-1}(x_1 - m_1), V_{22} - V_{21}V_{11}^{-1}V_{12}) \quad (5.3)
$$

Likewise, in our example, this would equate to saying to writing the distribution of $z$ in terms of its partitions as

$$
\begin{pmatrix}
  z_1 \\
  z_2
\end{pmatrix}
\sim
\mathcal{N}(0, \frac{1}{\lambda_z} \begin{pmatrix}
  \Sigma_{11} & \Sigma_{12} \\
  \Sigma_{21} & \Sigma_{22}
\end{pmatrix})
$$

Then, following the pattern expressed in equation (5.3), we write the conditional distribution of $z_2$ given $z_1$ as

$$
z_2|z_1 \sim \mathcal{N}(\Sigma_{21}\Sigma_{11}^{-1}z_1, \frac{1}{\lambda_z}(\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})) \quad (5.4)
$$

An intuitive way to look at the procedure we have employed is to think of $z_1$ as the elements of the vector that form a “basis” to describe the spatial process, and the $z_2$ elements as being so highly correlated with these “basis” elements that, computationally, they are completely determined by the $z_1$ elements. If that is the case, then we should be able to determine $z_2$ given $z_1$; that is, the variance in equation (5.4) should be zero, yielding

$$
z_2|z_1 \sim \mathcal{N}(\Sigma_{21}\Sigma_{11}^{-1}z_1, 0) \quad (5.5)
$$
To show this, we can use the spectral theorem (Strang, 1988), which states that any real symmetric matrix $A$ can be factorized according to

$$ A = Q\Lambda Q^T $$

where $\Lambda$ is a diagonal matrix with the eigenvalues on the diagonal and $Q$ is made of up orthonormal columns. The matrix $\Sigma$ is real and symmetric so we can also write $\Sigma$ as

$$ \Sigma = Q\Lambda Q^T $$ (5.6)

Further, we note that $\Sigma$ is $n \times n$, but because it is semi-positive definite, it has $r < n$ positive eigenvalues and $n - r$ zero-valued eigenvalues. Using the partitioning scheme previously described, we can group together all the columns with non-zero eigenvalues, and rewrite equation (5.6) using this grouping.

$$
\begin{pmatrix}
\Sigma_{11} & \Sigma_{12} \\
\Sigma_{21} & \Sigma_{22}
\end{pmatrix} =
\begin{pmatrix}
Q_1 & Q_2
\end{pmatrix}
\begin{pmatrix}
\Lambda_{11} & 0 \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
Q_1^T \\
Q_2^T
\end{pmatrix}
$$

where $Q_2$, with dimensions $n \times (n - r)$, denotes the columns that correspond to the zero-valued eigenvalues and $Q_1$, with dimensions $n \times r$, represents the other, “basis” columns. With simplification, this becomes

$$
\begin{pmatrix}
\Sigma_{11} & \Sigma_{12} \\
\Sigma_{21} & \Sigma_{22}
\end{pmatrix} =
Q_1 \Lambda_{11} Q_1^T
$$

so that only the orthonormal columns that correspond to the positive eigenvalues are remaining. Now, we further partition $Q_1$, representing the first $r$ rows (across all $r$ columns) as $Q_{11}$. The other $n - r$ rows (also across all $r$ columns) are denoted by $Q_{21}$.

$$
\begin{pmatrix}
\Sigma_{11} & \Sigma_{12} \\
\Sigma_{21} & \Sigma_{22}
\end{pmatrix} =
\begin{pmatrix}
Q_{11} \\
Q_{21}
\end{pmatrix}
\Lambda_{11}
\begin{pmatrix}
Q_{11}^T & Q_{21}^T
\end{pmatrix}
$$
After performing simple matrix multiplication, we have

$$
\begin{pmatrix}
\Sigma_{11} & \Sigma_{12} \\
\Sigma_{21} & \Sigma_{22}
\end{pmatrix}
= 
\begin{pmatrix}
Q_{11}A_{11}Q_{11}^T & Q_{11}A_{11}Q_{21}^T \\
Q_{21}A_{11}Q_{11}^T & Q_{21}A_{11}Q_{21}^T
\end{pmatrix}
$$

Remember that we are trying to show that the variance of the conditional distribution of $z_2$ given $z_1$, given by $\Sigma_{22} - \Sigma_{21}^{-1}\Sigma_{12}$, is zero. We substitute the partitions of the matrix on the right side of the equation for the partitions of $\Sigma$.

$$
\Sigma_{22} - \Sigma_{21}^{-1}\Sigma_{12} = 0
$$

$$
\Sigma_{22} - (Q_{21}A_{11}Q_{11}^T)(Q_{11}A_{11}Q_{11}^T)^{-1}\Sigma_{12} = 0
$$

$$
\Sigma_{22} - Q_{21}A_{11}Q_{11}^TQ_{11}^{-1}A_{11}^{-1}Q_{11}^{-1}\Sigma_{12} = 0
$$

$$
\Sigma_{22} - Q_{21}A_{11}^{-1}Q_{11}^{-1}\Sigma_{12} = 0
$$

$$
\Sigma_{22} - Q_{21}Q_{11}^{-1}\Sigma_{12} = 0
$$

$$
\Sigma_{22} - Q_{21}Q_{11}^{-1}Q_{11}A_{11}Q_{21}^T = 0
$$

$$
\Sigma_{22} - Q_{21}A_{11}Q_{21}^T = 0
$$

$$
Q_{21}A_{11}Q_{21}^T - Q_{21}A_{11}Q_{21}^T = 0
$$

So, we see that in the semi-positive definite case, the conditional distribution of $z_2$ given $z_1$ has variance equal to zero. This means that once we obtain $z_1$, we obtain $z_2$ using the mean of the conditional distribution given by equation (5.4), which is simply $\Sigma_{21}\Sigma_{11}^{-1}z_1$.

### 5.7.2 Full conditional distribution for $z_1$

Now that we have shown that the $z_2$ partition of the $z$ process is completely determined given $z_1$, $\Sigma_{11}^{-1}$, and $\Sigma_{21}$, we need to determine how to update $z_1$ using the values of the other parameters in our formulation. Recall that previously, we generated a realization of the entire $z$ vector, conditional on the parameters $\mu$, $\lambda_y$, $\lambda_z$, and $\Sigma$.

$$
z|\Sigma, \lambda_y, \mu, y, \lambda_z \sim N(\lambda_y(\lambda_yD^TD + \lambda_z\Sigma^{-1})^{-1}D^T(y - \mu), (\lambda_yD^TD + \lambda_z\Sigma^{-1})^{-1})
$$
This distribution was based on the likelihood for the model, given in equation (2.2) and the prior for $z$, given in equation (2.3). This is the distribution of the entire $z$ vector, requiring us to be able to obtain the matrix $\Sigma^{-1}$. We are interested in using the data and the other parameters to update only the "basis" elements of $z$, that is, the partition $z_1$, and we would then later determine the $z_2$ elements. So, we need to re-write the likelihood and prior in terms of $z_1$ to derive the full conditional in our new terms.

Let’s first look at the prior for $z$. We take a look at the original prior, written in terms of the various partitions previously discussed.

$$
\left( \begin{array}{c} z_1 \\ z_2 \end{array} \right) | \lambda_z, \Sigma \sim N \left( 0, \frac{1}{\lambda_z} \left( \begin{array}{cc} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{array} \right) \right)
$$

We know that the marginal distribution of a multivariate normal distribution is also normal, with the partitions that correspond to those elements. So, we can see that the marginal distribution of $z_1$ is

$$
z_1 | \lambda_z, \Sigma \sim N(0, \frac{1}{\lambda_z} \Sigma_{11})
$$

(5.7)

Next, we consider the likelihood, which is given by

$$
y \sim N(X \beta + Dz, \frac{1}{\lambda_y} I)
$$

We want to have this expression in terms of $z_1$ rather than $z$. Since we can determine the $z_2$ partition using only the fact that $z_2 = \Sigma_{21} \Sigma_{11}^{-1} z_1$. So, we can obtain the vector $z$ from the vector $z_1$ by using the construction

$$
z = \left( \begin{array}{c} I \\ \Sigma_{21} \Sigma_{11}^{-1} \end{array} \right) z_1
$$
Now, we substitute this formulation into the likelihood, allowing $\mathbf{A} = \Sigma_{21}\Sigma_{11}^{-1}$ for notational convenience.

$$y \sim N(X\beta + DAz_1, \frac{1}{\lambda_y}I)$$

To get the full conditional distribution for $z_1$, we combine the prior and likelihood, as was done before in the case of $z$. We obtain the following multivariate normal full conditional for $z_1$ based on the data and the other parameters:

$$z_1 | y, \mu, \lambda_y, \lambda_2, \Sigma \sim N(\lambda_y(\lambda_y A^T D^T DA + \lambda_2 \Sigma_{11}^{-1})^{-1} A^T D^T (y - \mu I), \lambda_y A^T D^T DA + \lambda_2 \Sigma_{11}^{-1})^{-1})$$

So, this new updating strategy allows us to overcome the obstacle of having a singular $\Sigma$ matrix (due to semi-positive definiteness). Instead of having to invert $\Sigma$, we only need to invert $\Sigma_{11}$, which is the subset of $\Sigma$ that is positive definite (and therefore of full rank). By using the appropriate partitions of $\Sigma$, $z$, and $y$, we can update first update $z_1$, based on the data and the values of the other parameters; then we use the relationship $z_2 = \Sigma_{21}\Sigma_{11}^{-1}z_1$ to obtain the values for $z_2$. Upon re-integrating these vectors, we again obtain $z$ in its original form.

### 5.7.3 Predicting $z'$ using $z_1$

Given that we followed all the previous steps correctly, we now have obtained the complete vector $z$, which gives the value of the spatial process for each of the $m$ points at which observations were made. However, this does not address the important issue of inferring the value of the spatial process at the points for which we have no data. We will designate these $p$ values as $z'$. Note that obtaining these values is of great practical concern. In most cases, not all the points of interest can be sampled, leading experts to try to construct a picture of the variable of interest over the whole region using whatever data is available.
In section 4.3.3, we discussed a procedure for sampling from the distribution of \( z' | z \). Basically, this involved generating from the conditional multivariate normal distribution as given in (4.4):

\[
z' | z \sim N(\Sigma'\Sigma^{-1}z, \Sigma'' - \Sigma'\Sigma^{-1}\Sigma'T)
\]

Here, \( \Sigma' \) and \( \Sigma'' \) are only partitions of the complete covariance matrix \( \Sigma_{total} \) needed in the joint distribution of \( (z, z') \) as given by (4.3).

\[
\Sigma_{total} = \begin{pmatrix} \Sigma & \Sigma'T \\ \Sigma' & \Sigma'' \end{pmatrix}
\]

However, to calculate the parameters of the conditional multivariate normal distribution \( z' | z \), we need the inverse of the correlation matrix, \( \Sigma^{-1} \). As we have discussed, this is not possible if the matrix \( \Sigma \) is semi-positive definite. The obvious alternative is generate \( z' \) conditional on \( z_1 \) instead of \( z \). Since we have already determined that \( z_2 \) depends completely on \( z_1 \), there is no reason to incorporate \( z_2 \) in our update for \( z' \).

Recall that the marginal distribution for \( z_1 \) given in (5.7) was

\[
z_1 | \lambda_z, \Sigma \sim N(0, \frac{1}{\lambda_z}\Sigma_{11})
\]

This yields the conditional distribution \( z' | z_1 \) as follows

\[
z' | z_1 \sim N(\Sigma'y'\Sigma_{11}^{-1}z_1, \Sigma'' - \Sigma'y'\Sigma_{11}^{-1}\Sigma'T)
\]

where \( \Sigma'y' \) is the partition of the covariance matrix \( \Sigma_{total} \) that consists of the rows corresponding to \( z_1 \) and the columns corresponding to \( z' \) and \( \Sigma' \) is its transpose. Note that this distribution is very similar to that given for \( z' | z \) in (4.4), only differing in the partitions of \( \Sigma_{total} \) that are used. So, \( z' \) can be easily obtained in the case of semi-positive definite \( \Sigma \), without obtaining \( \Sigma^{-1} \).
Figure 5.5: The real trend surrounded by the sample data

5.8 A simple example

Given the information in the preceding sections, we examine how this algorithm works in a small example. Consider the data set shown in Figure 5.5, which includes measurements taken along a straight path, given by the x-axis in this picture. The line gives an artificial trend of 201 points which was generated from a multivariate normal distribution with a mean of 2.0 and a covariance matrix given by

\[
\text{Cov}(z_i, z_j) = \frac{1}{\lambda_z} \Sigma
\]

\[
= \frac{1}{2} \exp\left\{-\frac{d_{ij}^2}{30^2}\right\}
\]

(5.8)

where \(d_{ij}\) is the distance between elements \(i\) and \(j\). The 50 points surrounding this line are selected points from the trend, with i.i.d. random noise added. We would like to recreate the trend by using this sample data, which is roughly of a quarter of
the original data points. To do this, we fit the model

\[ y = \mu \mathbf{1} + z + \varepsilon \]

![Graph of a spatial process](image)

**Figure 5.6:** The posterior means of the estimated spatial process

The methodology of this model has been discussed in chapter 4, and we use that approach here. However, we substitute the covariance matrix given above for that given by the process convolution approach. Since some of the sample points lie close together on the x-axis and since the scale parameter, 30, is fairly large given the range of this data, we can expect to have some numerical problems when attempting to take the Cholesky decomposition and the inverse of \( \Sigma \). It is not possible to denote precisely which points or how many points are causing this problem. The scaling parameter, given by \( \tau \) in equation 5.8, changes during the computation, as different values are proposed and accepted in the Metropolis-Hastings updating scheme. With each change, the covariance matrix changes also, leading to a constantly varying rank
for the matrix.

The dotted line in Figure 5.6 shows the posterior means of the spatial process (with the posterior mean of the trend $\mu$ added). The triangles mark the posterior means for the spatial process at which observations were taken. We can see that these are connected smoothly by the dotted line, giving a close representation of the original trend. To see how close our posterior estimates are to the real trend, refer to Figure 5.7. Once again, the real trend is given by the solid line; the estimated posterior mean trend is given by the dotted line. Of course, the data points and the triangles (denoting posterior mean estimates at places where observations were made) are also present for reference. We can see that the fitted trend is very close to the real trend, although it is somewhat more susceptible to the influence of the noise factor in the vicinity of the minimum and maximum points of the curve.

![Diagram showing posterior means of estimated trend, data points, and real trend](image)

**Figure 5.7:** The posterior means of the estimated trend, together with the sample data and the real trend
Chapter 6

Example: Application to the Piazza Road Superfund Site

6.1 Introduction to the Piazza Road data set

The Piazza Road dataset is taken from an actual pilot study of an EPA (U.S. Environmental Protection Agency) Superfund site in Missouri ((Ryti, 1993) and (Ryti et al., 1992)). The region was so designated due to widespread dioxin pollution from contaminated oil, which had been applied in an attempt to minimize dust storms in the area. A large area adjacent to Piazza Road was affected, and these measurements of soil contamination were taken from a small 100 feet × 200 feet region of that site during a pilot study that attempted to discover the structure of the dioxin distribution. Earlier studies showed that water runoff channels in the sites were polluted and gave reason to suspect that the pollutant had been spread through these means. This data set provides an example for our use in exploring spatial models and testing various spatial estimation methods.

Our data set was obtained from a stratified random sample. The site was divided into four quadrants, and fifty samples were taken uniformly from each quadrant. At each site, a one foot square area was measured out, and nine squares, of sixteen square
inches each, were delineated. A soil sample one tablespoon in size was taken from each small square, and all nine from the same site mixed together in a tin to form the sample from that particular site. The laboratory took three different measurements from each of the two hundred tins. For this reason, we expect some variability from the mixing of the soil in each tin and from various sources of laboratory error. Since the measurements were made in parts dioxin per billion, it seems very reasonable to transform using the natural logarithm. In further discussion of the data, we will assume that a given measurement is in natural logarithm of parts dioxin per billion.

Due to the large amount of data taken from the site, we have narrowed our focus down to only 120 measurements taken at various points in the site. Use of the entire data set is possible, but our goal here is to illustrate the use of the model and to see if a fairly small subset of the data can be used to give us an accurate picture of the non-stationarity at this site. The observation sites represented in our data set were chosen by selecting 60 observation points at random from each half of the site. It should be noted that our data set is somewhat restricted in that it does not include repeated observations at any particular observation site. The levels of contamination can be seen in Figure 6.1, in units log parts per billion.

Although we do not have extensive information about the geographical features of the landscape, we do know about one peculiarity of the site. In lower left-hand side of Figure 6.1, we see a roughly comma-shaped region in which the observed levels of dioxin contamination are noticeably higher. This area corresponds to a streambed at the Superfund site. As previously mentioned, it has been hypothesized that contamination is higher here because of the drainage of groundwater into natural drainage channels in the landscape.

Besides this dioxin transport along the stream channel, there are other factors that lead us to believe that a spatial process should be a feature of any model used
Figure 6.1: Measured levels of contamination for chosen points at the Piazza Rd. site
to analyze the Piazza Road data. For instance, we can expect that the pollutant was probably not applied uniformly to the road, but that there was some variation in the amounts applied as the applicator mechanism(s) ranged over the area. Afterwards, it is certain that various environmental forces, such as wind and rainfall, played a part in the present distribution of dioxin in the area. Even if we had this sort of information, it would be difficult to include the effects of these forces as parameters in the overall trend portion (expressed as $X\beta$) of the model. So, in this case, we have decided to make use of the model given by equation (2.1), which was

$$ y = X\beta + Dz + \varepsilon $$

However, since we only have one observation per site, the matrix $D$ is simply the identity matrix. In addition, we have no information that could be used as covariates to fit a trend of the form $X\beta$, so we substitute an overall mean, given as $\mu 1$.

$$ y = \mu 1 + z + \varepsilon $$

In the following sections we discuss various approaches that could be used to model the covariance structure of the spatial process responsible for the distribution of dioxin across the Piazza Road site.

### 6.2 Analysis using a simple Gaussian covariance model

One commonly employed and relatively simple analysis of the data can be made if we are willing to assume that the spatial process is isotropic in nature, meaning that the correlation between points only depends on the Euclidean distance between them. This can be implemented using a simple Gaussian covariance structure, where the correlation between any two points $s_i$ and $s_j$ is given by

$$ Corr(s_i, s_j) = \exp\left\{-\frac{d_{ij}^2}{\tau^2}\right\} $$  \hspace{1cm} (6.1)

81
Our prior for \( z \) would then be

\[ z \sim N(0, \frac{1}{\lambda_z} \Sigma) \]

which was earlier given by equation (2.3). Once again, \( \Sigma \) serves as the correlation matrix, with each element \( \Sigma_{ij} \) determined by the formula given in (6.1). This Gaussian covariance structure would then substitute for the process convolution approach which was discussed earlier in detail.

This model for the covariance structure has its own conveniences, as well as drawbacks. First, it has only two parameters governing it, \( \lambda_z \) and \( \tau \), making calculations fewer and simpler. Second, interpretation of the results may be easier. This formulation of covariance structure is more widely known and simple to interpret. However, this formulation assumes stationarity, so that the correlation between the spatial process at any two observed points at any points with the same relation in space anywhere within the region must be the same. In this example, we have doubts as to whether this assumption is valid, particularly since local features in the landscape likely affected the transport of the contaminant. For instance, the streambed in the lower left portion of the figure has large concentrations of dioxin, and we may question whether the spatial covariance here is different than that at other regions of the site due to the drainage of groundwater through this channel.

### 6.2.1 Differences in implementation

Notice in equation (6.1), that the parameter \( \tau \) has been given a new role. In this formulation, \( \tau \) is a scaling factor. The higher the value of \( \tau \), the larger the distance \( d_{ij} \) between two points \( i \) and \( j \) must be before the correlation of the spatial process between two points becomes negligible. In our implementation of this model, we make
the substitution $\theta = \frac{1}{\tau^2}$ so that we have

$$\text{Corr}(s_i, s_j) = \exp\{d_{ij}^2 \theta\}$$

avoiding some confusion with parameter designations.

The change in covariance formulation does not affect the update steps for $\mu$, $\lambda_y$, or $\lambda_z$, since none of these affects the computation of $\Sigma$. It also eliminates many of the steps in our MCMC procedure that were discussed in chapter 4, since the determination of $\Sigma$ does not rely on values for $\psi_x$, $\psi_y$, or the coordinate-axis scaling $\tau$. However, we must assign to the parameter $\theta$ a prior distribution, and also implement an updating procedure.

Since $\theta$ must be a positive parameter, a likely choice is the gamma distribution, given by

$$\pi(\theta|a_\theta, b_\theta) \sim \Gamma(a_\theta, b_\theta)$$

If no prior information or intuition is possessed about the likely range of values for $\theta$, $a_\theta = 1.0$ and $b_\theta = 0.005$ provide a nearly flat distribution for $\theta$. We chose this route in our example.

In order to find the full conditional distribution for $\theta$, we need the likelihood for the model, which is given in equation (2.6) as

$$L(y|\mu, \lambda_y, \lambda_z, \theta) \propto |S|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(y - \mu 1)^T S(y - \mu 1)\right)$$

where the precision matrix $S$ depends on the parameters $\lambda_y$, $\lambda_z$, and $\Sigma$ through (2.7):

$$S = \lambda_y [I - D(\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1} \lambda_y D^T]$$

Combining this likelihood with the prior leads us to the following full conditional distribution for $\theta$:

$$\pi(\theta|y, \mu, \lambda_y, \lambda_z) \propto |S|^{\frac{1}{2}} \theta^{a-1} \exp\left(-\frac{1}{2}(y - \mu 1)^T S(y - \mu 1) - b_\theta \theta\right)$$

83
Since this distribution is not of known form, we update $\theta$ with a Metropolis-Hastings step, with a proposal mechanism similar to that used for $\lambda_y$ and $\lambda_z$ in chapter 4.

### 6.2.2 Results

Figure 6.2 shows the posterior mean of the spatial process $z'$ over an evenly-spaced grid. Notable features include the area in the lower left side of the figure, which show the concentrations of dioxin to be substantially higher in the area surrounding the streambed. This is consistent with the drainage patterns of the landscape and with the data observed near this region. Due to the Gaussian covariance structure we have chosen, these predictions change rather smoothly over the area of interest.

### 6.3 Analysis using a process convolution approach

In this section, we address various implementations of the convolutions approach to modeling spatial covariance. Each of these approaches allows some degree of non-stationarity. The first approach allows the kernels at each point in the region to have varying orientation and eccentricity. The second approach extends the former to allow each kernel to have its own parameter that controls the shrinking/stretching of the coordinate axes, making it more flexible, but also more computationally intensive. We show how each approach can be used with the Piazza Road data, then we compare the results gained through both analyses and discuss the benefits of each approach.

#### 6.3.1 An approach restricting all kernels to the same size

In chapter 3, we described an approach in which the kernels were allowed to vary in eccentricity and orientation (these were measured through the parameters $\psi_x$ and $\psi_y$), as well as in size (measured through the shrinking/stretching parameter $\tau$). In this section, however, we discuss a more limited approach. We allow the $\psi_x$ and $\psi_y$ to
Figure 6.2:  Posterior mean surface using Gaussian covariance structure
vary smoothly for kernels across the region, but we maintain the same \( \tau \) for all points in the region. This is equivalent to assuming that the \( \tau \) parameter approaches \( \infty \), so that the elements of the \( \tau \) vector are completely dependent on one another. As \( \tau \) approaches \( \infty \), then, regardless of the distance \( d_{ij} \) between any two points \( i \) and \( j \) in the region, the correlation of the spatial process between these points approaches one. Then, \( \tau \) is the same for each kernel throughout the region; non-stationarity will only be evident in the shapes of the kernels.

This approach has advantages over the full approach described in chapter 3. Since all the elements of the vector \( \tau \) are held to the same value, there are many fewer parameters to estimate. Instead of the \( m \) elements of \( \tau \), we only have to deal with one \( \tau \) value, which is then duplicated for the \( m - 1 \) others. Also, without spatial dependence between the elements of \( \tau \) to take into account, the correlation matrix for the multivariate normal distribution of \( \tau \) and its governing parameter, \( \tau \), are not necessary. This results in great savings in programatic record-keeping, computational resources, and time.

Without a differing \( \tau \) vector, we lose some flexibility. We cannot allow for the fact that kernels of the same shape and rotation may have differing levels of influence on the spatial process measured at other points. One purpose of our analyses here is to discover whether, for this example, adding these parameters seems to make a large difference in the result, or if the advantage presented by these parameters is not large enough to justify the extra resources needed to include them.

**Implementation issues**

For updating purposes, we consider only a single scalar value. Without the whole vector \( \tau \), the structure of the model is greatly simplified, since we no longer need to consider the spatial dependence among the \( \tau \) elements. Previously, we were driven
toward the multivariate normal prior for the vector \( \tau \), because of the ability of this distribution to simply express the covariance among a set of parameters. Now that this no longer necessary, we switch to a more natural gamma prior on \( \frac{1}{\sqrt{\tau}} \). This transformation \( \theta = \frac{1}{\sqrt{\tau}} \) is on a similar scale to a precision parameter, for which a gamma prior is also a very popular choice. Therefore, we express the lone element \( \tau \) on a different scale, and set the prior and updating steps accordingly. After each MCMC update for \( \theta \), we set each element of \( \tau \) equal to \( \frac{1}{\theta \tau} \), and continue with the updates for \( \mu, \lambda_y, \lambda_z \), and \( \tau_\psi \) as discussed in chapter 4.

As just mentioned, for the prior distribution of \( \theta \) we choose a gamma prior distribution with parameters \( a_\theta \) and \( b_\theta \). Similarly to the Gaussian case described in the previous section, our full conditional distribution for the \( \theta \) can then be written

\[
\pi(\theta|y, \mu, \lambda_y, \lambda_z) \propto |S|^{1/2} \theta^{a_\theta - 1} \exp\left(-\frac{1}{2}(y - \mu \mathbf{1})^T S(y - \mu \mathbf{1}) - b_\theta \theta \right)
\]

where

\[
S = \lambda_y [\mathbf{I} - D(\lambda_y D^T D + \lambda_z \Sigma^{-1})^{-1} \lambda_y D^T]
\]

A large part of the computation time to execute a Metropolis-Hastings step in this case will be spent in recalculating the matrix \( \Sigma \) for the candidate value of \( \theta \) (using also the current values for \( \psi_x \) and \( \psi_y \)), and from this, recalculating the matrix \( S \). However, since there is only one value to update, instead of a whole vector \( \tau \), this uses considerably less time than the full model discussed in chapter 3.

Results

Figure 6.3 shows the posterior mean of \( \mu \mathbf{1} + z \). This image is produced with the same range of shading as Figure 6.2, to facilitate comparisons. We can see once again that the major center of pollution is found in the lower left portion of the graph, with much lower concentrations along the length of the right side.
Figure 6.3: Posterior mean surface using changing orientation ellipses
However, Figure 6.3 does not allow us to get any assessment of the amount of non-stationarity that is occurring in the region; this was one of the original purposes behind this approach to the data. Getting an impression about this element of the analysis is more difficult. In Figure 6.4, we see a plot featuring representations of the ellipses, with the ellipses blown up by a factor of two to make them easier to see. Remember that each ellipse represents the bivariate normal kernel used at that location; these ellipses are determined by $\psi_x$, $\psi_y$, and $\theta$ (which is used to coordinate the shrinking/expansion of the axes at all locations). The ellipses drawn with solid lines are at locations for which we have observed data; those drawn with dotted lines are at the grid points for which we are trying to make predictions about the spatial process $z$.

When drawing the ellipses, one area of potential concern is how to determine what the posterior mean of the ellipses should be. We are calculating the ellipses based on $\psi_x$ and $\psi_y$, but if we just average the the x-coordinates of the foci for each ellipse and the y-coordinates of the foci for each ellipse we may not be getting the most accurate picture. This is because there are two combinations of foci coordinates that will draw the same ellipse; these are $(\psi_x, \psi_y)$ and $(-\psi_x, -\psi_y)$. If we average these, the resulting focus may be entirely misleading, maybe giving us a focus that is much closer to the center $(0, 0)$ than is actually the case.

One solution to this problem, which we have made use of in Figure 6.4 and later figures, is to average the ellipses radially. Each ellipse can be drawn using a set of polar coordinates, with each coordinate pair consisting of the angle and the distance from the origin to the ellipse at that angle. Imagine one particular location in the region, for which we have obtained, through MCMC, an set of focus coordinates $(\psi_x, \psi_y)$. For each iteration’s focus, we use the coordinates $(\psi_x, \psi_y)$ to calculate a set of polar coordinates that defines the ellipse in that iteration. After this process
Figure 6.4: Orientation-varying ellipses averaged radially
has been completed for each iteration, we look at all the distances that have been calculated for each angle, and average them. The posterior mean ellipse, then, is defined for each location in this manner. However, this is somewhat of a misnomer, because after the posterior averaging, the shape that we obtain may not precisely be an ellipse, but this difference is generally not too large.

We can combine all this information into one graph that shows the posterior mean of the contaminant over a grid of points in the region, as well as the radially averaged posterior mean ellipses. Figure 6.5 overlays Figure 6.3 with Figure 6.4. This plot enables us to grasp the finer details about the covariance structure in conjunction with the pattern of pollution.

Some information about the spatial dependence structure is available through the ellipses. The non-stationary elements of the covariance structure are concentrated in areas where there is a trend of heavy contamination, and they are oriented in along this trend. In the lower left corner of the figure, we can see that the contamination level (measured in log parts per billion) is quite high in comparison with the rest of the site. This corresponds with the streambed which we know is present in the geography of the region. Along the course of the streambed, we can see a pronounced elongation of the ellipses, curving in the direction of the stream’s course. The curvature traced out by the ellipses is particularly strong along the edges of the streambed; this is most noticeable in the lower left corner and in the the middle of the plot (near the right hand side of the most heavily polluted area). This makes sense, as we would expect the measure of dioxin in the stream to be most dependent on the the amount of contamination directly upstream or downstream and less dependent across the stream’s current. In the right-hand side of Figure 6.5, we can see the ellipses are almost circular. This means that in this region, the spatial dependence is almost isotropic, so that all pairs of points separated by the same distance have the same
Figure 6.5: Posterior mean surface using same-area ellipses
Figure 6.6: 95% credible intervals for the fitted values plotted with observations (using same-area ellipses)

correlation.

In summary, Figure 6.6 shows the 95% credible intervals obtained using this implementation. These intervals are plotted in a sorted order according to the posterior mean of the fitted value \((\mu_1 + z)\) for the log concentration of dioxin. Along with each interval, we have plotted the corresponding observation. We notice that none of the observed values are far from the fitted range, so that our model seems to be fitted the data reasonably well. We see a cluster of intervals and points at the left side due to the fact that there were a large number of very low observed concentration levels. Sensibly, the fitted values were similarly low, and we see a clustering in the lower left hand corner. In fact, over twenty observations shared the same low observed level that was the minimum observed level in the data set.
6.3.2 Full approach, allowing size of kernels to vary across locations

Implementation issues

We now address our results using the full model discussed in Chapter 3. This model uses the vectors $\psi_x$ and $\psi_y$ to allow the ellipses to vary in orientation and eccentricity, as well as allowing for a shrinking/expansion vector $\tau$. This parameterization allows us significantly more flexibility than we had in the previous case, in which we assumed that all ellipses were the same size. The kernel at each location now has a corresponding value of $\tau$, which enables it to effectively reduce or expand the range of its influence on the spatial process. Note that the parameters of each ellipse are still calculated according to a constant area (in our case, this value is 3.5), but now the scaling factor $\tau^2$ multiplies the the variance matrix of the normal kernel, as given in equation (3.5). So, this change in $\tau$ does not affect our correspondence in shape and orientation between the covariance matrix for the kernel and ellipse. The ellipses have simply been expanded by some multiple of $\tau$ so that they have relative areas which depict how the kernel is scaled at that location.

Note that in our implementation, the results of which are given in the next section, we have not placed prior distributions on the parameters $\lambda_{\tau}$ and $\tau_{\tau}$. (These parameters determine the covariance matrix for $\tau$, given our assumption that the dependence among the elements of $\tau$ has a Gaussian covariance structure.) Instead, we have set these values at $\lambda_{\tau} = 0.1$ and $\tau_{\tau} = 50$, which we think are reasonable since we want the sizes of the kernels to vary smoothly across the region. Through much experimentation, we have determined that in order to include these two parameters in the MCMC update scheme, we need very informative priors. Otherwise, there is a tendency for the updating scheme to follow the data too closely. More details are given in our discussion about possible extension of the work in Chapter 7.
Results

The posterior mean of $\mu_1 + z$ is portrayed in Figure 6.7, which is once again produced with the same range of shadings as Figures 6.3 and 6.2. This image is almost identical to the ones which preceded it, with the highest concentration of pollutants visible in the “hot spot” of the lower left side.

However, Figure 6.7 does not allow us to gain any understanding of the extent of non-stationary exhibited in this example. For a better understanding of the underlying covariance structure of the spatial process, we refer to Figure 6.8. Like Figure 6.4, this plot shows only the ellipses which correspond to the shape of the normal kernel centered at each point. The ellipses have been radially averaged, just as in the previous implementation.

Examining Figure 6.8 closely, we can see that in addition to their orientations, the sizes of the ellipses are changing very gradually across the plot. Although this may not be noticeable at a first glance, there is a general trend of ellipses increasing in size as we move from left to right across the region. This pattern is most noticeable in the upper right corner and the lower right corner of the plot. It seems that the ellipses are larger in the areas in which the level of contamination is low and nearly level for long distances. It is reasonable that the ellipses should be larger in this area; the response varies only slightly, so that the spatial process at these points is highly correlated. However, the orientations of the ellipses remains similar to our implementation of the model without allowing for the size variations.

In Figure 6.9, we see the image plot of Figure 6.7 overlaid by Figure 6.8. This representation allows for easier comparison between the extent of contamination at any group of locations and the extent and nature of non-stationarity in that area. In this case, we can see that, once again, the non-stationarity is most noticeable along the boundaries of the streambed in the lower left portion of the figure. This may
Figure 6.7: Posterior mean surface using ellipses that change in both orientation and size
Figure 6.8: Orientation and size-varying ellipses averaged radially
be due to the fact that the contamination levels increase rapidly over a fairly small distance from the boundary of the stream into its channel. The transport mechanism in this area seems to be much more determined by the direction of the water flow here, whereas outside this area it is generally more isotropic in nature.

In Figure 6.10, the 95% credible intervals obtained using this implementation are displayed. Once again, these intervals are plotted in a sorted order according to the posterior mean of the fitted value for the log concentration of dioxin; the observed data that corresponds to each interval is also shown. As before, our model seems to be fitted the data reasonably well, although the credible intervals are smaller using this analysis (compared to Figure 6.6). In the next section, we compare these two implementations more closely.

Comparison of the process convolution implementations

One point of interest may be whether the more complicated model (in which the elements of $\tau$ are not constrained to be the same) actually provides a better estimate of the spatial process at most locations than the simpler method (in which the elements of $\tau$ are identical). One way to look at examine this question is to compare estimates of the spatial process, and examine their variance or examine the posterior intervals associated with these estimates.

We choose four locations from various sections of the region, and we compare the 95% posterior intervals for the spatial process at these points. Figure 6.11 shows the locations of these four points, overlaid on the image background of Figure 6.7. The 95% posterior interval for each point is given in Figure 6.12. The number below each set of intervals corresponds to its location as numbered in Figure 6.11. The intervals drawn with a solid line are those obtained by the more flexible approach, allowing the kernels to vary in size across the site. The dotted segments delineate the intervals
Figure 6.9: Posterior mean surface using size-varying ellipses
Figure 6.10: 95% credible intervals for the fitted values plotted with observations (using size-varying ellipses)

obtained using the more restrictive model, which maintains the same size for each kernel.

We can see that in each case, the more flexible approach (solid lines) corresponds to a shorter posterior interval than is given by the more restrictive approach (dotted lines). Although the difference is not always large, the posterior distribution of the spatial process at most locations has less variability under the varying $\tau$ approach. In some examples, this difference may be more extreme; in others, the difference may be negligible.
Figure 6.11: Location of the four points chosen for comparison
Figure 6.12: Posterior intervals for four chosen points at the Piazza Road site.
Chapter 7

Conclusions and possible extensions

In previous chapters, we have discussed the methodology of our approach to non-stationary modeling and the results gained from this analysis. This chapter seeks to conclude by summarizing the main points of the dissertation and by detailing likely possibilities for future research.

We have shown that the process convolution approach is one feasible way to allow for non-stationarity in covariance structure. It allows the data to help us determine this structure, and allows us to incorporate the uncertainty associated with this structure fairly easily. This, in particular, separates it from many other approaches to covariance modeling, in which the uncertainty associated with the choice of the model can be difficult to quantify. Our portrayal of the set of kernels in terms of their one standard deviation ellipses can be evaluated by those interested in the patterns of dependency among various locations of the region.

Because of our decision to define each of the bivariate normal kernels in terms of their corresponding ellipses, we have made our parameterization based on the defining geometrical properties of the ellipse. In particular, we chose to use the focus and area of the ellipses, along with their centers at each observation point, to define the kernels. However, we could just as easily have used other geometrical properties,
such as eccentricity or angle of rotation. Although these geometrical properties can be derived from one another, this approach is not limited to the parameterization we have selected; others are possible and may be more feasible in some situations.

Another place for improvement is the prior specification for $\tau,$ a vector which controls the scaling of each of the kernels. In Chapter 3, we specified the form of the prior for $\tau$ as

$$\tau \sim N(\mu_{\tau}, 1, \frac{1}{\lambda_{\tau}}\Sigma_{\tau})$$

Prior distributions were also placed on the hyperparameters $\lambda_{\tau}$ and $\tau_{\tau}$ in Chapter 4, which together complete the specification of the Gaussian covariance structure.

$$\tau_{\tau} \sim \Gamma(\alpha_{\tau}, \beta_{\tau})$$
$$\lambda_{\tau} \sim \Gamma(\alpha_{\lambda_{\tau}}, b_{\lambda_{\tau}})$$

However, we were not able to update these last two parameters as part of our MCMC updating scheme; instead, we fixed the parameters at what seemed to be appropriate values for our application. It would be more desirable to allow both $\lambda_{\tau}$ and $\tau_{\tau}$ to be determined by the data and their respective prior distributions. As we discovered through experimentation, though, this is not a trivial task. A more informative set of priors would be of great assistance. If priors used are not very informative, a tendency exists for the model to fit each observation as i.i.d. noise.

It could be that an alternative parameterization of $\tau$ is more appropriate. For instance, since each $\tau_i$ must be positive, we might try to model $\tau$ according to a lognormal multivariate distribution, yielding

$$\log(\tau) \sim N(\mu_{\tau}, 1, \frac{1}{\lambda_{\tau}}\Sigma_{\tau})$$

Although slightly more complicated to implement, this model may be more intuitive for a parameter which must always remain positive.
Figure 7.1: Using disks to define a kernel

One potential weakness of the methodology that we have proposed is that it is not efficient with large amounts of data, say on the order of thousands of observations. The matrix manipulations required by our current methodology may not be feasible for such a large number of data points. In some applications, it is not unusual to obtain data sets on the order of thousands of observations over hundreds or thousands of unique locations. One potential solution may be to specify a set of “basis” ellipses, and then estimate the weights that are necessary to specify the right kernel for each site using these “basis” elements. A fast Fourier transform could then be done to obtain the spatial realization.

An area in which further work is being performed involves the use non-parametric kernels (Kern and Higdon, 1999). We can think of these as a stack of concentric disks
stacked over each location, with varying radii at each point, as in Figure 7.1. This work also seeks to address the difficulties of “edge effects” in regions where there is discontinuity in the spatial field. For instance, if water temperatures are the quantity of interest, any island or isthmus that interrupts the normal current pattern will also present serious difficulties in the estimation of the spatial process in adjoining areas. One goal of this research is to allow alternative metrics to calculate distance around the obstacles disturbing the flow. This work proposes a new distance metric which would define paths as appropriate around or over any unusual feature in the region.
Bibliography


Press, William H., Flannery, Brian P., Teukolsky, Saul A. and Vetterling, William T.


Biography

I am a native of Newport News, Virginia. I entered the Massachusetts Institute of Technology in the fall of 1991 and obtained the S.B. degree in mathematics (with applied emphasis) in May 1994. I began my studies at Duke University in August 1994, and received the M.S. degree in statistics in the spring of 1996. I spent the next 14 months working as an applications developer at SAS Institute in Cary, North Carolina, and re-entered the university in the fall of 1997. My professional goals include further research into Bayesian spatial models and statistical computing, with an emphasis on teaching.