

Hierarchical Bayesian spatio–temporal models for population spread

Christopher K. Wikle and Mevin B. Hooten

There is a long history in the ecological sciences concerning the development of mathematical models for describing the distribution of organisms over time and space. Although such models have sometimes been evaluated by comparison to observations, they have seldom been “fit” to data in a formal statistical sense. Thus, uncertainties regarding data, model, and parameters are not readily accounted for in such analyses. Alternatively, realistic statistical models for spatio–temporal processes in ecology often require that one estimate a very large number of parameters. This is typically not possible given the relatively limited amount of data collected over space and time. Critically, such a statistical model could be simplified (in terms of reducing the number of parameters) by accommodating the well known empirical and theoretical results concerning the process (e.g. partial differential equations (PDEs) or integro–difference equations (IDEs) for population spread). That is, the statistical model can make use of the well established mathematical models for the process. The Bayesian hierarchical paradigm for spatio–temporal models allows one to account for the aforementioned sources of uncertainty and yet still include such prior knowledge for the process, parameters and measurements. In this chapter, we provide an introduction to process-based hierarchical Bayesian spatio–temporal models for population spread, focusing primarily on PDE dynamics. We illustrate the concepts and demonstrate the methodology on the problem of predicting the Eurasian Collared-Dove invasion of North America.

8.1 Introduction

The spread of populations has long been of interest to ecologists and mathematicians. Whether it be the invasion of gypsy moths in North America, soybean rust in Southern Africa and South America, avian influenza in Asia, or seemingly countless other invasive species and emerging diseases, it is clear that the invasion of ecosystems by exotic organisms is a serious concern. Given the increasing economic, environmental, and human health impact of such invasions, it is imperative that in addition to understanding the basic ecology of such processes, we must be able to monitor them in near real time, and to combine that data and our basic ecological understanding to forecast, in space and time, the likely

spread of the population of interest. Perhaps more importantly, we must be able to characterize realistically and account for various types of uncertainty in such forecasts.

For sure, the dynamics of population spread are complicated. The underlying processes are potentially nonlinear, nonhomogeneous in space and/or time, related to exogenous factors in the environment (e.g. weather), and dependent on other competitive species. Ecologists have long been interested in these issues (e.g. Elton 1958). Traditionally, the modeling of such processes has been motivated by applied mathematicians and the use of PDEs, IDEs, and discrete time–space models (e.g. Hastings 1996). The differences in these models are primarily related

to whether one wishes to consider time and/or space discrete or continuous. Although there are fundamental differences in these approaches, from a theoretical limiting perspective, there are notions of equivalence between them. From a practical perspective, in the presence of data, some sort of discretization in time and/or space is typically necessary, whether it be in the form of finite differences, finite elements, or spectral expansions.

The modeling approaches described above have most often been used to form “theoretical predictions,” usually in the form of calculating the theoretical velocity of the dispersive wave front for the population of interest. Ecologists have calculated the average velocity of spread given observations and compared such estimates to the theoretical spread (e.g. Andow et al. 1990; Caswell 2001). Although a useful endeavor in order to provide understanding of the basic utility of theoretical (often deterministic) models, several limitations are apparent in this approach with regard to “operational” prediction over diverse habitats. One concern is that in order to get analytical solutions to the PDE or IDE models, substantial simplifications in the dynamics must be made. For instance, in the PDE case, an assumption of homogeneous diffusion and/or net reproductive rate is typical. For IDE models, the redistribution kernels that are necessary for analytical solution may not be representative of the data, and the assumption of homogeneity of the kernels over space and time may be unrealistic. Perhaps more critically, in general there have been only a few attempts to actually fit these theoretical models to data in a statistically rigorous fashion. Part of the reason for this is the traditional lack of relatively complete, high resolution spatio-temporal ecological data. Even when available, the data for such processes are typically assumed to be known without error. In practice, there is a great deal of sampling and measurement error in observations of ecological processes that when unaccounted for results in misleading analyses.

There is increasing recognition that new methods for spatio-temporal processes that efficiently accommodate data, theory, and the uncertainties in both must be developed (Clark et al. 2001). The hierarchical Bayesian approach is ideal for this as it allows one to specify uncertainty in components of the problem

conditionally, ultimately linked together via formal probability rules (see Wikle 2003a for an overview). This framework explicitly accepts prior understanding, whether that be from previous studies, or ecological theory (e.g. Wikle 2003b). Furthermore, it easily accommodates multiple data sources with errors and potentially different resolutions in space and time (e.g. Wikle et al. 2001). Finally, complicated dependence structures in the parameters that control the population dynamics can be accommodated quite readily in the hierarchical Bayes approach (e.g. Wikle et al. 1998; Wikle 2003b).

Although hierarchical Bayesian models for spatio-temporal dynamical problems such as population spread are relatively easy to specify, there are a number of complicating issues. First and foremost is the issue of computation. Hierarchical Bayesian models are most often implemented with Markov Chain Monte Carlo (MCMC) methods. Such methods are very computationally intensive, especially in the presence of complicated spatio-temporal dependence and large prediction/sampling networks. The issue of high dimensionality, in the sense of a very large number of parameters in the model, is especially important in spatio-temporal models. It is critical that one be able to efficiently parameterize the dynamical process in such models. As with any model building paradigm, there are also potential issues of model selection and validation.

In this chapter we seek to illustrate, through a simplified example, how one can use the hierarchical Bayesian methodology to develop a model for the spread of the Eurasian Collared-Dove. This model will consider data, model and parameter uncertainty. The dynamical portion of the model will be based on a relatively simple underlying diffusion PDE with spatially varying diffusion coefficients. Section 8.2 will describe the statistical approach to modeling spatio-temporal dynamic models. Section 8.3 then describes schematically the hierarchical Bayesian approach to spatio-temporal modeling. Next, Section 8.4 contains the Eurasian Collared-Dove invasion case study and the associated hierarchical Bayesian model. Section 8.5 contains a discussion and suggestion for an alternative reaction-diffusion model, and finally, Section 8.6 gives a brief summary and conclusion.

8.2 Statistical spatio-temporal dynamic models

Assume we have some spatio-temporal process $Y(s; t)$, where s is a spatial location in some spatial domain D (typically in two-dimensional Euclidean space, but not restricted to that case) and t denotes time, $t = \{t_1, \dots, t_T\}$. Most processes in the physical, environmental and ecological sciences behave in such a way that the process at the current time is related to the process at a previous time (or times). We refer to such a process as a *dynamical process*. Given that such processes cannot be completely described by deterministic rules, it would be ideal to characterize the joint distribution of this process for all times and spatial locations. Typically, this is not possible without some significant restrictions on the distribution. A common restriction is to assume the process behaves in a Markovian fashion; that is, the process at the current time, conditioned on all of the past, can be expressed completely by conditioning only on the most recent past. For example, consider the case where we have a finite number of spatial locations $\{s_1, \dots, s_n\}$ and discrete times $t = \{0, 1, 2, \dots, T\}$. Let $\mathbf{Y}_t \equiv (Y(s_1; t), \dots, Y(s_n; t))'$, where we use the prime to denote a vector or matrix transpose. Then, the joint distribution of the spatio-temporal process can be factored as follows:

$$\begin{aligned} [\mathbf{Y}_0, \dots, \mathbf{Y}_T] &= [\mathbf{Y}_T | \mathbf{Y}_{T-1}, \dots, \mathbf{Y}_0] \\ &\quad \times [\mathbf{Y}_{T-1} | \mathbf{Y}_{T-2}, \dots, \mathbf{Y}_0] \dots \\ &\quad \times [\mathbf{Y}_2 | \mathbf{Y}_1, \mathbf{Y}_0] [\mathbf{Y}_1 | \mathbf{Y}_0] [\mathbf{Y}_0], \end{aligned} \quad (8.1)$$

where we use the brackets $[\]$ to denote distribution and $[a|b]$ to denote the conditional distribution of a given b . With the first-order Markov assumption, (8.1) can be written,

$$\begin{aligned} [\mathbf{Y}_0, \dots, \mathbf{Y}_T] &= [\mathbf{Y}_T | \mathbf{Y}_{T-1}] [\mathbf{Y}_{T-1} | \mathbf{Y}_{T-2}] \dots \\ &\quad \times [\mathbf{Y}_2 | \mathbf{Y}_1] [\mathbf{Y}_1 | \mathbf{Y}_0] [\mathbf{Y}_0]. \end{aligned} \quad (8.2)$$

This Markovian assumption is a dramatic simplification of (8.1), yet one that is very often realistic for dynamical processes. From a modeling perspective, we then must specify the component distributions $[\mathbf{Y}_t | \mathbf{Y}_{t-1}]$, $t = 1, \dots, T$. In general, we write this in terms of some function $\mathbf{Y}_t = f(\mathbf{Y}_{t-1}; \boldsymbol{\theta})$, where the parameters $\boldsymbol{\theta}$ describe the dynamics of the process. This function can be nonlinear, and the associated

distribution can be Gaussian or nonGaussian. For illustration, consider the first-order linear evolution equation with Gaussian errors,

$$\mathbf{Y}_t = \mathbf{H}\mathbf{Y}_{t-1} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_\eta), \quad (8.3)$$

where the “propagator” or “transition” matrix \mathbf{H} is an $n \times n$ matrix of typically unknown parameters. Consider the i th element of \mathbf{Y}_t and the associated evolution equation implied by (8.3),

$$Y(s_i; t) = \sum_{k=1}^n h(i, k) Y(s_k; t-1) + \eta(s_i; t), \quad (8.4)$$

where $h(i, k)$ refers to the element in the i -th row and k -th column of \mathbf{H} . Thus, (8.4) shows that the process value at location s_i at time t is a linear combination of all the process values at the previous time, with the relative contribution given by the “redistribution” weights $h(i, k)$, and the addition of possibly correlated noise $\eta(s_i; t)$.

In the statistics literature, the model (8.3) is known as a first order vector autoregressive (VAR(1)) model (e.g. see Shumway and Stoffer 2000). Such models are easily extended to higher order time lags and more complicated error processes.

8.2.1 Simple example

As a simple example, for $n = 3$ spatial locations, we need to specify the relationship between $Y_t(s_i)$ and $Y_{t-1}(s_1)$, $Y_{t-1}(s_2)$, $Y_{t-1}(s_3)$, for each $i = 1, \dots, 3$. Consider the linear relationship:

$$\begin{bmatrix} Y_t(s_1) \\ Y_t(s_2) \\ Y_t(s_3) \end{bmatrix} = \begin{bmatrix} h_{11}Y_{t-1}(s_1) + h_{12}Y_{t-1}(s_2) \\ \quad + h_{13}Y_{t-1}(s_3) + \eta_t(s_1) \\ h_{21}Y_{t-1}(s_1) + h_{22}Y_{t-1}(s_2) \\ \quad + h_{23}Y_{t-1}(s_3) + \eta_t(s_2) \\ h_{31}Y_{t-1}(s_1) + h_{32}Y_{t-1}(s_2) \\ \quad + h_{33}Y_{t-1}(s_3) + \eta_t(s_3) \end{bmatrix} \quad (8.5)$$

or

$$\begin{bmatrix} Y_t(s_1) \\ Y_t(s_2) \\ Y_t(s_3) \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} Y_{t-1}(s_1) \\ Y_{t-1}(s_2) \\ Y_{t-1}(s_3) \end{bmatrix} + \begin{bmatrix} \eta_t(s_1) \\ \eta_t(s_2) \\ \eta_t(s_3) \end{bmatrix}, \quad (8.6)$$

where the weights $h_{ik} \equiv h(i, k)$ describe how the process at location k at the previous time influences the location i at the current time. We have also added a contemporaneous noise process $\eta_i(s_i)$ to “force” the system.

8.2.2 Parameterization

The difficulty with such formulations in practice is that for most environmental and ecological processes the number of spatial locations of interest, n , is quite large, and there is simply not enough information to obtain reliable estimates of all parameters $h(i, k), i, k = 1, \dots, n$. Thus, we typically must parameterize the propagator matrix \mathbf{H} in terms of some parameters θ , whose dimensionality is significantly less than the n^2 required to estimate \mathbf{H} directly.

Perhaps the simplest statistical parameterization for \mathbf{H} is to assume $\mathbf{H} = \mathbf{I}$, a multivariate random walk. Although advantageous from the perspective of having the fewest (0) parameters in \mathbf{H} , this model is nonstationary in time. More importantly, such a structure is not able to capture complex interaction across space and time, and is not realistic for most physical, environmental, and ecological processes. A natural modification is to allow $\mathbf{H} = \text{diag}(\mathbf{h})$, a diagonal matrix with elements on the diagonal potentially varying with spatial location. Such a model is nonseparable in space–time, yet it still does not account for realistic interactions between multiple spatial locations across time.

Below, we consider two alternative, yet related, approaches for parameterizing \mathbf{H} .

8.2.3 IDE-based dynamics

To capture dynamical interactions in space–time that are realistic for ecological processes, the propagator matrix \mathbf{H} must contain nonzero off-diagonal elements. This can be seen clearly from the IDE perspective. Consider the linear stochastic IDE equation,

$$Y(s; t) = \int k(s, r)Y(r; t-1)dr + \eta(s; t), \quad (8.7)$$

where the error process $\eta(s; t)$ is correlated in space, but not time, and the redistribution kernel $k(s, r)$ describes how the process at the previous time is redistributed to the current time. Although similar

to equation (8.4), the IDE equation considers continuous space rather than discrete space. General IDE equations are quite powerful for describing ecological processes (e.g. Kot et al. 1996); the dynamics are controlled by the properties of the redistribution kernel. For example, the dilation of the kernel controls the rate of diffusion, and advection can be controlled by the skewness of the kernel (Wikle 2002). In addition, the characteristics of the dynamics that can be explained are affected by the kernel tail thickness and modality. Although such models are rich in describing complicated ecological processes, they have not often been “fit” to data in a rigorous statistical framework. Wikle (2002) and Xu et al. (2005) show that such models can be fit to data and that allowing the kernels to vary with spatial location can dramatically increase the complexity of the dynamics modeled. From our perspective, a discretization of (8.7) suggests potential parameterizations of \mathbf{H} as a function of the kernel parameters, θ . Such parameterizations include nonzero off-diagonal elements, and can be nonsymmetric (i.e. $h(i, k) \neq h(k, i)$) allowing for complicated interactions in time and space while using relatively few kernel parameters.

Disadvantages of using IDE models in this setting are related to the implementation within a statistical framework, parameter estimation (although hierarchical Bayes approaches help), choice of an appropriate kernel, accommodating spatially varying parameters, and reduced computational efficiency due to potentially nonsparse \mathbf{H} matrices.

8.2.4 PDE-based dynamics

The IDE-based dynamics of the previous section suggest that the simplest, realistic statistical parameterization of \mathbf{H} would have diagonal and nonsymmetric nondiagonal elements. One could simply parameterize such a model statistically (e.g. see Wikle et al. 1998). However, in the case of physical and ecological processes, we often know quite a bit about the theory of the underlying dynamical process through differential equations (e.g. see Holmes et al. 1994). In the case of linear PDEs, standard finite differencing implies equations such as (8.3). More importantly, such discretizations imply parameterizations of \mathbf{H} in terms of important parameters of

the PDE, as well as the finite-difference discretization parameters (e.g. Wikle 2003b).

Consider the general diffusion PDE,

$$\frac{\partial u}{\partial t} = \mathcal{H}(u, w, \theta), \quad (8.8)$$

where \mathcal{H} is some function of the variable of interest, u , other potential variables, w , and parameters θ . Simple finite difference representations (e.g. see Haberman 1987) suggest an approximate difference equation model,

$$\mathbf{u}_t = h(\mathbf{u}_{t-\Delta_t}, \mathbf{w}, \theta) + \boldsymbol{\eta}_t, \quad (8.9)$$

where we have added the noise term $\boldsymbol{\eta}_t$ to account for the error of discretization. Note, it is also reasonable to consider this error term to be representative of model errors in the sense that the PDE itself is an approximation of the real process of interest.

Now, for illustration, consider the simple diffusion equation,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(\delta(x, y) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\delta(x, y) \frac{\partial u}{\partial y} \right), \quad (8.10)$$

where $u_t(x, y)$ is a spatio-temporal process at spatial location $\mathbf{s} = (x, y)$ in two-dimensional Euclidean space at time t and $\delta(x, y)$ is a spatially varying diffusion coefficient. Forward differences in time and centered differences in space (e.g. see Haberman 1987) give the difference associated with equation (8.10),

$$\begin{aligned} u_t(x, y) &= u_{t-\Delta_t}(x, y) \left[1 - 2\delta(x, y) \left(\frac{\Delta_t}{\Delta_x^2} + \frac{\Delta_t}{\Delta_y^2} \right) \right] \\ &+ u_{t-\Delta_t}(x - \Delta_x, y) \left[\frac{\Delta_t}{\Delta_x^2} \{ \delta(x, y) \right. \\ &\quad \left. - (\delta(x + \Delta_x, y) - \delta(x - \Delta_x, y))/4 \right] \\ &+ u_{t-\Delta_t}(x + \Delta_x, y) \left[\frac{\Delta_t}{\Delta_x^2} \{ \delta(x, y) \right. \\ &\quad \left. + (\delta(x + \Delta_x, y) - \delta(x - \Delta_x, y))/4 \right] \end{aligned}$$

$$\begin{aligned} &+ u_{t-\Delta_t}(x, y + \Delta_y) \left[\frac{\Delta_t}{\Delta_y^2} \{ \delta(x, y) \right. \\ &\quad \left. + (\delta(x, y + \Delta_y) - \delta(x, y - \Delta_y))/4 \right] \\ &+ u_{t-\Delta_t}(x, y - \Delta_y) \left[\frac{\Delta_t}{\Delta_y^2} \{ \delta(x, y) \right. \\ &\quad \left. - (\delta(x, y + \Delta_y) - \delta(x, y - \Delta_y))/4 \right] \\ &+ \eta_t(x, y), \end{aligned} \quad (8.11)$$

where it is assumed that the discrete u -process is on a rectangular grid with spacing Δ_x and Δ_y in the longitudinal and latitudinal directions, respectively, and with time spacing Δ_t . Again, the error term $\eta_t(x, y)$ has been added to (8.11) to account for the uncertainties due to the discretization as well as other model misspecifications.

From (8.11) it can be seen that the discretization can be written as (8.4) or (8.3) where the propagator (redistribution) matrix \mathbf{H} depends upon the diffusion coefficients $\boldsymbol{\delta} = [\delta(s_1), \dots, \delta(s_n)]'$ and the discretization parameters Δ_t , Δ_x , and Δ_y ,

$$\begin{aligned} \mathbf{u}_t &= \mathbf{H}(\boldsymbol{\delta}, \Delta_t, \Delta_x, \Delta_y) \mathbf{u}_{t-\Delta_t} \\ &+ \mathbf{H}_B(\boldsymbol{\delta}, \Delta_t, \Delta_x, \Delta_y) \mathbf{u}_{t-\Delta_t}^B + \boldsymbol{\eta}_t, \end{aligned} \quad (8.12)$$

where again, \mathbf{u}_t corresponds to an arbitrary vectorization of the gridded u -process at time t , $\mathbf{H}(\boldsymbol{\delta}, \Delta_t, \Delta_x, \Delta_y)$ is a sparse $n \times n$ matrix with essentially five nonzero diagonals corresponding to the bracket coefficients in (8.11), hence its dependence on $\boldsymbol{\delta}$. Note also that we have included a separate boundary specification in that $\mathbf{u}_{t-\Delta_t}^B$ is an $n_B \times 1$ vector of boundary values for the u -process at time $t - \Delta_t$, and $\mathbf{H}_B(\boldsymbol{\delta}, \Delta_t, \Delta_x, \Delta_y)$ is an $n \times n_B$ sparse matrix with elements corresponding to the appropriate coefficients from (8.11). Thus, the product $\mathbf{H}_B(\boldsymbol{\delta}, \Delta_t, \Delta_x, \Delta_y) \mathbf{u}_{t-\Delta_t}^B$ is simply the specification of model edge effects.

8.2.5 Simple example

Expanding on the previous simple example, consider the three equally spaced (i.e. Δ_x is constant) spatial locations (in 1D space) x_1, \dots, x_3 and boundary points x_0 and x_4 . Assuming for ease of notation

that $\Delta_t = 1$ we then can write the dynamical portion of (8.12) as:

$$\begin{bmatrix} u_t(x_1) \\ u_t(x_2) \\ u_t(x_3) \end{bmatrix} = \begin{bmatrix} \theta_1(x_1)u_{t-1}(x_1) + \theta_2(x_1)u_{t-1}(x_2) \\ + \theta_3(x_1)u_{t-1}(x_0) \\ \theta_1(x_2)u_{t-1}(x_2) + \theta_2(x_2)u_{t-1}(x_3) \\ + \theta_3(x_2)u_{t-1}(x_1) \\ \theta_1(x_3)u_{t-1}(x_3) + \theta_2(x_3)u_{t-1}(x_4) \\ + \theta_3(x_3)u_{t-1}(x_2) \end{bmatrix} \quad (8.13)$$

where for $i = 1, 2, 3$,

$$\begin{aligned} \theta_1(x_i) &= 1 - \delta(x_i) \left(\frac{\Delta_t}{\Delta_x^2} \right), \\ \theta_2(x_i) &= \frac{\Delta_t}{\Delta_x^2} \{ \delta(x_i) + (\delta(x_{i+1}) - \delta(x_{i-1}))/4 \}, \\ \theta_3(x_i) &= \frac{\Delta_t}{\Delta_x^2} \{ \delta(x_i) - (\delta(x_{i+1}) - \delta(x_{i-1}))/4 \}. \end{aligned}$$

This can then be written,

$$\begin{aligned} \begin{bmatrix} u_t(x_1) \\ u_t(x_2) \\ u_t(x_3) \end{bmatrix} &= \begin{bmatrix} \theta_1(x_1) & \theta_2(x_2) & 0 \\ \theta_3(x_2) & \theta_1(x_2) & \theta_2(x_2) \\ 0 & \theta_3(x_3) & \theta_1(x_3) \end{bmatrix} \\ &\times \begin{bmatrix} u_{t-1}(x_1) \\ u_{t-1}(x_2) \\ u_{t-1}(x_3) \end{bmatrix} + \begin{bmatrix} \theta_3(x_1) & 0 \\ 0 & 0 \\ 0 & \theta_2(x_3) \end{bmatrix} \\ &\times \begin{bmatrix} u_{t-1}(x_0) \\ u_{t-1}(x_4) \end{bmatrix} \end{aligned} \quad (8.14)$$

which is, in matrix form,

$$\mathbf{u}_t = \mathbf{H}(\delta, \Delta_t, \Delta_x) \mathbf{u}_{t-1} + \mathbf{H}_B(\delta, \Delta_t, \Delta_x) \mathbf{u}_{t-1}^B. \quad (8.15)$$

8.2.6 Population growth

The basic diffusion model (8.10) is quite powerful in that the diffusion coefficients are allowed to vary with space, which is appropriate for landscape-scale modeling since diffusion rates are dependent upon many spatially varying factors. However, this model does not include a growth term and thus the process $u_t(x, y)$ decays over time. A more realistic PDE for

many ecological processes that exhibit population growth is given by a reaction–diffusion equation,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(\delta(x, y) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\delta(x, y) \frac{\partial u}{\partial y} \right) + f(u), \quad (8.16)$$

where in addition to the diffusive terms in (8.10) we have added the “reaction” term $f(u)$ that describes the population growth dynamics. The classic reaction–diffusion equation was originally discussed by Fisher (1937) and Skellam (1951), and gives diffusion plus logistic population growth,

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left(\delta(x, y) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\delta(x, y) \frac{\partial u}{\partial y} \right) \\ &+ \gamma_0 u \left(1 - \frac{u}{\gamma_1} \right), \end{aligned} \quad (8.17)$$

where γ_0 is the intrinsic population growth rate and γ_1 is the carrying capacity. In vector form, (8.17) can be written,

$$\begin{aligned} \mathbf{u}_t &= \mathbf{H}(\delta, \Delta_t, \Delta_x, \Delta_y) \mathbf{u}_{t-\Delta_t} \\ &+ \mathbf{H}_B(\delta, \Delta_t, \Delta_x, \Delta_y) \mathbf{u}_{t-\Delta_t}^B \\ &+ \gamma_0 \mathbf{u}_{t-\Delta_t} - \gamma_0 \gamma_1 \text{diag}(\mathbf{u}_{t-\Delta_t}) \mathbf{u}_{t-\Delta_t} + \boldsymbol{\eta}_t, \end{aligned} \quad (8.18)$$

where the $\text{diag}(\)$ operator simply makes the vector argument a diagonal matrix with the argument along the main diagonal. Note that this model is nonlinear in the parameters γ_0 and γ_1 and in the process, $\mathbf{u}_{t-\Delta_t}$.

8.3 Hierarchical Bayesian models

As one might imagine, a key challenge to implementation of a model such as (8.12) or (8.18) is the estimation of the spatially varying diffusion coefficients, δ . From a classical statistical perspective, this would be very difficult for several reasons (e.g. simultaneous likelihood based estimation of δ and hence \mathbf{H} , also nonlinearity in (8.18)). However, from a hierarchical Bayesian perspective, such estimation is relatively easy. In this section, we give a very brief overview of the hierarchical approach, as general details can be found in modern Bayesian textbooks such as Gelman et al. (2004), and in overview papers such as Wikle (2003a) as well as other chapters in this volume.

8.3.1 Basic hierarchical modeling

Hierarchical modeling is based on a simple fact from probability that the joint distribution of a collection of random variables can be decomposed into a series of conditional models. For example, if a , b , c are random variables, then basic probability allows us to write the factorization $[a, b, c] = [a|b, c][b|c][c]$. In the case of spatio-temporal models, the joint distribution describes the behavior of the process at all spatial locations and all times. This is difficult to specify for complicated processes. Typically, it is much easier to specify the distribution of the conditional models. In that case, the product of the series of relatively simple conditional models gives a joint distribution that can be quite complex.

When modeling complicated processes in the presence of data, it is helpful to write the hierarchical model in three basic stages (Berliner 1996):

Stage 1. Data Model: [data | process, data parameters]

Stage 2. Process Model: [process | process parameters]

Stage 3. Parameter Model: [data and process parameters].

The basic idea is to approach the complex problem by breaking it into subproblems. Although hierarchical modeling has been around a long time in Statistics (e.g. see the Bibliographic note in Chapter 5 of Gelman et al. 2004), this basic formulation for modeling complicated temporal and spatio-temporal processes in the environmental sciences is relatively new (e.g. Berliner 1996; Wikle et al. 1998). The first stage is concerned with the observational process or “data model,” which specifies the distribution of the data given the fundamental process of interest and parameters that describe the data model. The second stage then describes the process, conditional on other process parameters. For example, in the diffusion model setting, the process stage would be factored in a Markovian sense as in (8.2), conditional on the spatially varying diffusion coefficients δ and the parameters that describe the noise process, η_t . Finally, the last stage models the uncertainty in the parameters, from both the data and process stages.

For example, we might model the diffusion coefficients in terms of spatially varying covariates and/or a spatially correlated random field. Note that each of these stages can have many substages (e.g. see Wikle et al. 1998; Wikle et al. 2001).

Our goal is to estimate the distribution of the process and parameters updated by the data. This posterior distribution is obtained via Bayes’ theorem:

$$\begin{aligned} & [\text{process, parameters}|\text{data}] \\ & \propto [\text{data}|\text{process, parameters}] \\ & \times [\text{process}|\text{parameters}][\text{parameters}]. \end{aligned} \quad (8.19)$$

Bayes’ theorem serves as the basis for Bayesian hierarchical modeling and when written in its general probability form (i.e. [posterior] \propto [likelihood] \times [prior]) we see that statistical conclusions are drawn from the “posterior” which is proportional to the data model (i.e. likelihood) times our *a priori* knowledge (i.e. the prior). Although simple in principle, the implementation of Bayes’ theorem for complicated models can be challenging. One challenge concerns the specification of the parameterized component distributions on the right hand side of (8.19). Although there has long been a debate in the Statistics community concerning the appropriateness of “subjective” specification of such distributions, such choices are a natural part of science based modeling. In fact, the incorporation of scientific information into these prior distributions provides a coherent mechanism by which one can incorporate the uncertainty related to these specifications explicitly in the model. Perhaps more importantly from a practical perspective is the calculation of the posterior. The complex and high-dimensional nature of ecological models (and indeed, most spatio-temporal models) prohibits the direct evaluation of the posterior. However, one can utilize Markov Chain Monte Carlo (MCMC) approaches to draw samples from the posterior distribution. Indeed, the use of MCMC for Bayesian hierarchical models has led to a revolution in that realistic (i.e. complicated) models can be considered in the analysis of spatio-temporal processes. Yet, we still typically have to formulate the conditional models in such problems with regard to the computational burden. Thus, the model building phase requires not only scientific understanding

of the problem, but must also address how that understanding can be modified to fit into the MCMC computational framework.

8.4 Eurasian Collared-Dove case study

The Eurasian Collared-Dove (*Streptopelia decaocto*) was first observed in the United States in the mid 1980s. This species originated in Asia and, starting in the 1930s, expanded its range into Europe (Hudson 1965). These birds were introduced into the Bahamas in 1974 from a population that escaped captivity (Smith 1987) and spread to the United States soon thereafter. Since its introduction in Florida, its range has been expanding dramatically across North America.

The outstanding success of the Eurasian Collared-Dove as an invader is well documented. It is less clear, however, precisely why it has been able to demonstrate such a significant range expansion. In their recent summary of the North American invasion, Romagosa and Labisky (2000) discuss the evidence that the birds show a remarkable ability for long-range dispersal, even in the presence of geographical barriers such as mountains and large bodies of water, and that the dispersing birds typically become successful breeders within two years. They speculate that possible factors for the range expansion are genetic, the ability to successfully adapt to human habitat, and a very high reproductive potential. It is widely believed that they will rapidly spread across North America much in the same way they spread across Europe (Romagosa and Labisky 2000).

In their study of the early expansion of the Eurasian Collared-Dove in Florida based on the Christmas Bird Count (CBC) data, Romagosa and Labisky (2000) found that the birds expanded northwesterly throughout the Florida peninsula and into the panhandle throughout the mid-1980s and early 1990s. They found that the expansion was most prevalent along the coasts, followed by “backfilling” into inland areas, consistent with a hypothesis of “jump” dispersal and population coalescence. They also found that since the dispersal from southern Florida occurred when the population abundance was low, dispersal was not likely density dependent.

Our goal with this case study is to consider the expansion of the Eurasian Collared-Dove in North America on the continental scale. The purpose of this analysis is to illustrate the Bayesian hierarchical methodology for incorporating partial differential equation priors in statistical spatio-temporal models. The data, hierarchical model, and results are described in the following subsections.

8.4.1 Data

Eurasian Collared-Dove data were obtained from the North American Breeding Bird Survey (BBS), and were collected by volunteer observers each breeding season along specified routes (Robbins et al. 1986). BBS sampling units are roadside routes of length approximately 39.2 km, along which an observer makes 50 stops and counts birds by sight and sound for a period of three minutes. There are over 4000 routes in the survey, but not all are sampled each year. Furthermore, there is a great deal of uncertainty in these observations, given the differences in experience and expertise of the volunteer observers (e.g. Sauer et al. 1994). In the case of the Eurasian Collared-Dove, this uncertainty is compounded by the fact that these birds look very similar to the Ringed Turtle-Dove. Although there are fundamental differences in the respective appearances and songs, it is thought that observers routinely mistake these species. This was probably even more the case early in the invasion, when observers had less experience distinguishing between the species.

We consider 18 years of BBS data, from 1986 through 2003. Figure 8.1 shows a plot of the counts at the sampled routes for each year. The circle size is proportional to the observed BBS count. Figure 8.2 shows the aggregated counts for each year. We consider these counts to be relative abundances since the probability of detection is not known. Nevertheless, these two plots show that there is clearly an invasion and the population is increasing exponentially with time.

8.4.2 Hierarchical model

This section describes a Bayesian hierarchical model for the invasion of the Eurasian Collared-Dove.

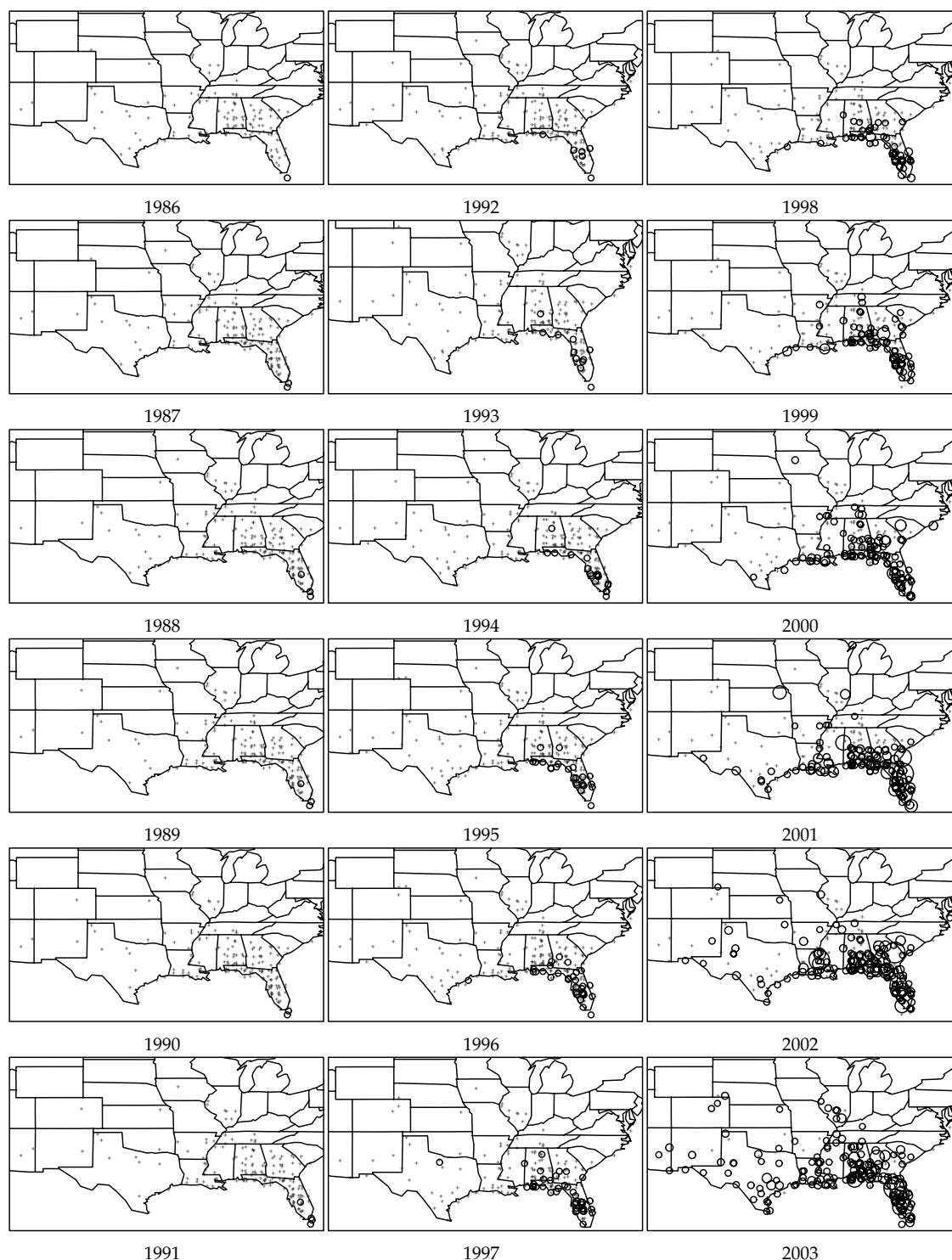


Figure 8.1. Location of BBS survey route (+) and observed Eurasian Collared-Dove count for years 1986–2003. The radius of the circles are proportional to the observed count.

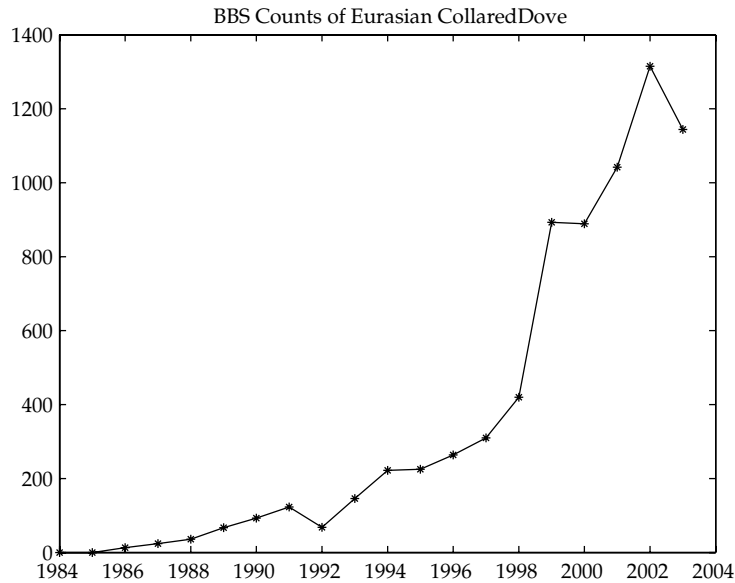


Figure 8.2. Sum of BBS Eurasian Collared-Dove counts over space for years 1986–2003.

The data model, process model, and parameter models are described in the following subsections. The results from the analysis are then presented, followed by a description of an alternative model.

8.4.2.1 Data model

For simplicity of illustration, we aggregate the observations on the grid shown in Figure 8.3. Specifically, we consider a lattice covering a portion of the continental United States (on an equal area projection). We let $N(s_i; t)$ correspond to the number of routes sampled in year t in grid box s_i . Then, $Z(s_i; t)$ corresponds to the total count in the i th grid box in year t over the $N(s_i; t)$ sampled routes. We denote the vector of counts over all grid boxes for year t by, $\mathbf{Z}_t = (Z(s_1; t), \dots, Z(s_n; t))'$. For purposes of maintaining the simplicity of this example, “missing” BBS routes were assumed to have zero counts, although more complex methods exist for dealing with such missing data. We then assume that conditional on an unknown spatio-temporal intensity process, the relative abundances are independent. Thus, we write

$$\mathbf{Z}_t | \boldsymbol{\lambda}_t \sim \text{Poi}(\text{diag}(\mathbf{N}_t) \boldsymbol{\lambda}_t), \quad t = 1, \dots, T, \quad (8.20)$$

where $\boldsymbol{\lambda}_t = (\lambda(s_1; t), \dots, \lambda(s_n; t))'$, $\mathbf{N}_t = (N(s_1; t), \dots, N(s_n; t))'$, and diag places the vector \mathbf{N}_t on the diagonal of an $n \times n$ matrix of zeros.

8.4.2.2 Process models

We now assume that the log of the Poisson intensity process is controlled by a latent (i.e. underlying) spatio-temporal process, $\mathbf{u}_t = (u(s_1; t), \dots, u(s_n; t))'$ plus independent noise,

$$\log(\boldsymbol{\lambda}_t) = \mathbf{u}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}) \quad (8.21)$$

or, equivalently,

$$\log(\boldsymbol{\lambda}_t) | \mathbf{u}_t, \sigma_\epsilon^2 \sim N(\mathbf{u}_t, \sigma_\epsilon^2 \mathbf{I}), \quad t = 1, \dots, T. \quad (8.22)$$

In this case, the error process $\boldsymbol{\epsilon}_t$ accounts for small-scale spatio-temporal variation (subgrid scale) and is independent across space and time. One could argue that it would be reasonable to allow this error process to be spatially correlated, yet for simplicity of illustration, we do not consider such correlation here.

Critical to the process modeling is the latent spatio-temporal process \mathbf{u}_t . Okubo (1986) showed that diffusion PDEs work well in modeling avian invasions. Thus, analogous to Wikle (2003b), we

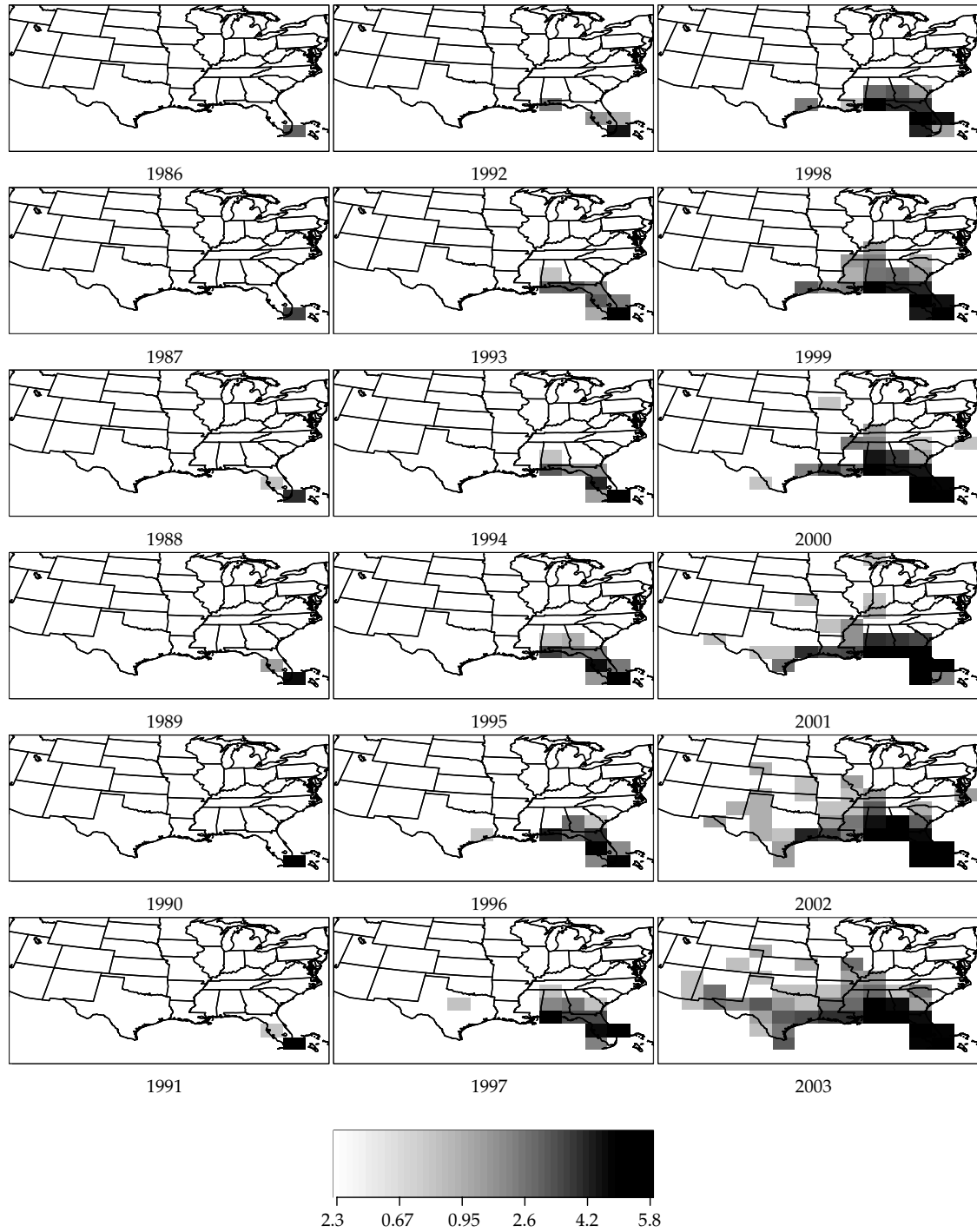


Figure 8.3. Log of Eurasian Collared-Dove BBS counts aggregated to a grid for years 1986–2003.

model this process via the discretized basic diffusion equation (8.12),

$$\mathbf{u}_t | \mathbf{u}_{t-1}, \delta \sim N(\mathbf{H}(\delta)\mathbf{u}_{t-1}, \sigma_\eta^2 \mathbf{R}_\eta), \quad (8.23)$$

where we have made several assumptions relative to (8.12). In particular, we assume $\Delta_t = 1$, $\Delta_x = \Delta_y = 1$ and the boundary process is zero everywhere (i.e. the grid locations outside of those shown in Figure 8.3 are defined to be zero for all time). As shown in Wikle et al. (2002), it is not difficult to allow the boundary process to be random within the hierarchical framework. However, the assumption of u taking the value zero on the boundaries is not unreasonable here, given that the boundaries correspond to ocean areas or areas of the domain in which the birds have not been observed yet. Although it could be argued that we should allow \mathbf{R}_η to contain spatial dependence, for simplicity of illustration, we let $\mathbf{R}_\eta = \mathbf{I}$ in this example. Furthermore, the Markovian structure in the u -process requires a specification of the initial condition \mathbf{u}_0 . We assign this a prior distribution, $\mathbf{u}_0 \sim N(\tilde{\mathbf{u}}_0, \Sigma_0)$. We let $\tilde{\mathbf{u}}_0 = \mathbf{0}$ and $\Sigma_0 = 10\mathbf{I}$, reflecting our vague belief in the initial process.

We note that the process models given by (8.22) and (8.23) are probably not the most realistic and are different from those given in Wikle (2003b) for modeling the spread of the House Finch over time. We choose this model because it is the simplest for illustrating the methodology of utilizing PDE priors in spatio-temporal hierarchical models. Wikle (2003b) considered an overall temporal trend term, modeled as a random walk in time. In addition, the diffusion equation considered in Wikle (2003b) included an exponential growth term. In the present example, we did not feel it appropriate to model the overall trend term as it is somewhat unrealistic. That is, the assumption of a common mean log intensity valid for all spatial locations at a given time is not realistic since there is definite spatial structure in the latent intensity, and most of the domain of interest has near zero intensity for most times, t . In the presence of data and with the added flexibility of the error term η_t in the basic diffusion equation, it is possible that the basic model considered here can accommodate the spread evident in the data. (Note, we discuss below in Section 8.5 a reaction-diffusion process model for \mathbf{u}_t that is more flexible in this regard.)

8.4.2.3 Parameter models

The primary parameters of interest here are the diffusion coefficients δ . A reasonable model for δ is given by,

$$\delta | \alpha, \sigma_\delta^2, \mathbf{R}_\delta \sim N(\Phi\alpha, \sigma_\delta^2 \mathbf{R}_\delta), \quad (8.24)$$

where Φ is an $n \times p$ known design matrix, α is a $p \times 1$ vector of “regression” coefficients, and the error has mean zero and is potentially spatially correlated with covariance matrix $\sigma_\delta^2 \mathbf{R}_\delta$. Ideally, one would include habitat covariates in Φ as suggested in Wikle (2003b). For example, for the Eurasian Collared-Dove we might include a human population covariate since this species is known to favor human-modified habitat. In that case, the error process could account for unknown habitat (or other) covariates that influence the spatial variation of the diffusion coefficients.

Alternatively, taking a simpler approach, we consider Φ to be the first p eigenvectors from a spatial correlation matrix (i.e. the so called empirical orthogonal functions, EOFs, which are simply space-time principal components). That is, we specify an $n \times n$ correlation matrix $\mathbf{R}(\theta)$ for the n grid locations, where the correlation function is positively definite and depends on spatial dependence parameter θ . We then get the symmetric decomposition $\mathbf{R}(\theta) = \Psi\Lambda\Psi'$ where Ψ is an $n \times n$ matrix of the eigenvectors of $\mathbf{R}(\theta)$ and Λ is a diagonal matrix of corresponding eigenvalues. The eigenvectors are orthogonal, so that $\Psi\Psi' = \Psi'\Psi = \mathbf{I}$. Typically, if the spatial dependence suggested by θ is fairly large, then most of the eigenvalues are very small and, as is usually the case with principal component analysis, one can retain most of the variability of the process described in $\mathbf{R}(\theta)$ by considering the largest $p \ll n$ eigenvalues/eigenvectors. Thus, we set Φ to be the $n \times p$ matrix of eigenvectors corresponding to the p largest eigenvectors of Ψ . Given that we are accounting for the potential spatial structure in δ through $\Phi\alpha$, we then set \mathbf{R}_δ equal to the identity matrix, \mathbf{I} . Essentially, we are modeling potential spatial structure in the δ field through the conditional mean (and hence α) rather than the covariance. This “trick” is to facilitate computation since the independent error structure and orthogonality ($\Phi'\Phi = \mathbf{I}$) simplifies the MCMC computations. A disadvantage of this approach is

that if spatial parameters were more explicitly modeled, posterior inference about the spatial structure could be made. Such computational tricks are probably not required here since the prediction grid is relatively small ($n = 111$), but for realistic grid sizes (densities) such computational considerations are critical.

In terms of the analysis presented herein, we based $\mathbf{R}(\theta)$ on the exponential correlation function, $r_\theta(d) = \exp(-\theta d)$, where d is a Euclidean distance between grid locations (e.g. d ranges from 0 to about 0.6 on our grid). We specify $\theta = 4$ (fixed) and keep $p = 8$ of the eigenvectors to start with (which account for about 80% of the variation). However, in this example, after preliminary analysis was performed, it was decided that only the first eigenvector was significantly influencing the analysis (i.e. p was reduced to 1). It would be relatively simple to allow θ to be a random parameter in this model corresponding to arbitrary spatial dependence, but for simplicity of illustration, it is fixed here.

A model for the regression coefficients α , is then

$$\alpha \sim N(\alpha_0, \sigma_\alpha^2 \mathbf{R}_\alpha), \tag{8.25}$$

where α_0 is the prior mean (specified to be a vector of zeros here) and \mathbf{R}_α corresponds to a known diagonal matrix with the p diagonal elements corresponding to the first p eigenvalues of \mathbf{A} , defined above.

We also must specify prior distributions for all of the variance parameters. For convenience, we give them all conjugate inverse gamma (IG) priors. That is,

$$\begin{aligned} \sigma_\epsilon^2 &\sim \text{IG}(q_\epsilon, r_\epsilon), & \sigma_\eta^2 &\sim \text{IG}(q_\eta, r_\eta), \\ \sigma_\delta^2 &\sim \text{IG}(q_\delta, r_\delta), & \sigma_\alpha^2 &\sim \text{IG}(q_\alpha, r_\alpha), \end{aligned} \tag{8.26}$$

where the q and r parameters are given (e.g., $q_\epsilon = q_\delta = q_\alpha = 2.8$, $r_\epsilon = r_\delta = r_\alpha = 0.28$, $q_\eta = 2.9$, $r_\eta = 0.175$), corresponding to relatively vague prior knowledge.

8.4.2.4 Implementation

The full-conditional distributions corresponding to the hierarchical model presented above are given in Appendix A. Furthermore, a sketch of the MCMC algorithm is presented, and R code is given. For the results presented here, the MCMC was run for

50,000 iterations, with the first 20,000 considered burn-in. Convergence was assessed subjectively by visual inspection of the sampling chains. Ultimately, MCMC output was resampled to mitigate autocorrelation in the chains.

8.4.3 Results

Figure 8.4 shows histograms of some of the variance parameters in the model.

The uncertainty in the posterior estimates of the spatially averaged Poisson intensity $\text{diag}(\mathbf{N}_t)\lambda_t$ is illustrated in Figure 8.5, which shows the 95% credible interval from the posterior. Figure 8.6 illustrates the uncertainty in the actual Poisson rate itself (i.e. λ_t) on the log scale.

Figure 8.7 shows the posterior mean of the spatial diffusion coefficient (δ) and Figure 8.8 shows the posterior standard deviation. Note that the posterior mean shows a few diffusion coefficients less than zero. Of course, this is not meaningful in terms of the original PDE, but is the model's attempt at adapting to the data in about the only way that it can. This is illustrated even more clearly when one considers predictions. Consider the $\log(N(s_i; t)\lambda(s_i; t))$ process. Figure 8.9 shows the posterior mean of the $\log(N(s_i; t)\lambda(s_i; t))$ process for each year. One can readily see the diffusion in this plot. The prediction of the Poisson intensity process (i.e. $\log(N(s_i; t)\lambda(s_i; t))$) for 2004 is shown in Figure 8.10 (assuming the number of routes sampled in each grid cell remains the same as in 2003). Note that the maximum intensity on the log scale (6.4) is larger in 2004 than in 2003 (5.8 on the log scale). At first glance one might wonder how the model can predict such growth given that there is no growth term in the prior model specification. We note that a condition for the model to be stationary is that the eigenvalues of \mathbf{H} must be less than 1 in modulus. The \mathbf{H} for this model that is built with the posterior mean of δ is nonstationary, as there are 5 eigenvalues that are greater than 1 in modulus. Thus, the model can exhibit explosive growth and predictions for 2004 are likely to grow quite large. Indeed, one assumes that many of the realizations of δ imply even larger eigenvalues for individual samples of \mathbf{H} and thus, the predictive distribution is unrealistically wide. Thus, our naive model with no growth term has

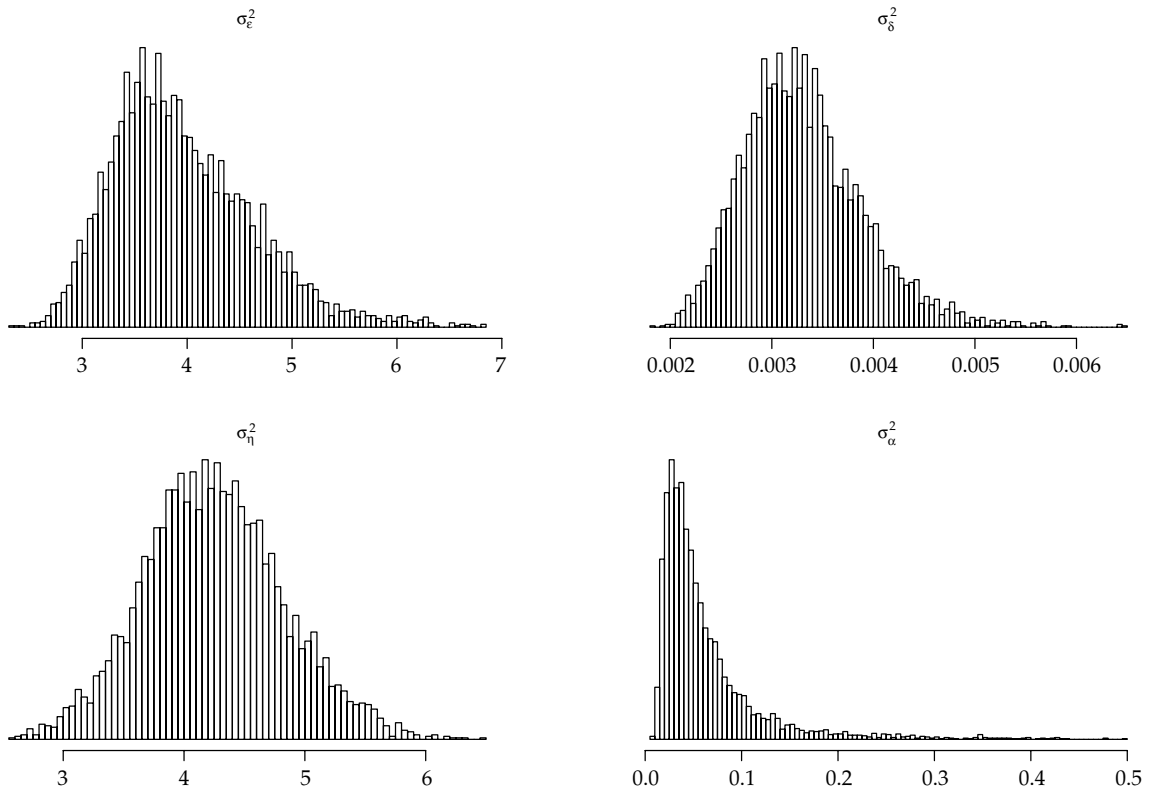


Figure 8.4. Histogram of samples from the posterior distribution of (a) σ_ϵ^2 , (b) σ_η^2 , (c) σ_δ^2 , and (d) σ_α^2 .

adapted to the data in the only way that it could, by choosing δ 's that imply explosive (i.e. exponential) growth.

8.5 Discussion

The similarity of Figures 8.2 and 8.5 is quite striking, but is to be expected in a strongly data-driven process. By considering this Poisson intensity to be random, we can associate some amount of uncertainty with this process (as evident in the credible interval). Furthermore, the Poisson rate itself (Figure 8.6) may be even more meaningful because this is the posterior mean Poisson intensity per sampled route over time. Thus the increase in intensity over time is indeed a result of the invasive species and not just an artifact of an increased sampling intensity over time.

The maps showing the posterior mean and standard deviation of δ (Figures 8.7 and 8.8) suggest

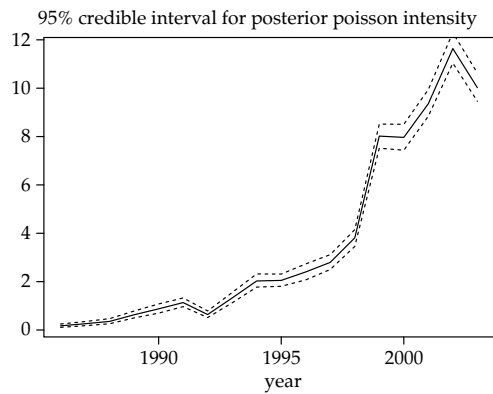


Figure 8.5. Credible interval for the posterior distribution of the Poisson intensity ($\text{diag}(N_t)\lambda_t$) averaged over space for years 1986–2003.

that although Eurasian Collared-Dove appears to be dispersing more readily in Louisiana and Mississippi, the variability associated with the mean estimates imply that the diffusion parameter

may not be significantly different over the spatial domain.

From a natural resources management perspective, the prediction for 2004 (Figure 8.10) is not encouraging. One main advantage of employing this model is that various types of uncertainty have been

accounted for and yet this model still suggests that the exponentially increasing population size and range expansion of the Eurasian Collared-Dove is indeed significant.

As mentioned above, the diffusion PDE selected for this case study is very simple, although still quite powerful with the spatially varying diffusion coefficients. A more plausible model would include some population growth term. For example, the reaction diffusion model given in (8.17) and (8.18) would be reasonable to consider. However, we note that (8.18) is nonlinear in \mathbf{u}_{t-1} and thus the full conditionals for $\mathbf{u}_t, t = 1, \dots, T$ cannot be derived in closed form. One could resort to Metropolis-Hastings sampling here, with for example, the linearized model as the proposal distribution. Metropolis-Hastings implementations in such high-dimensional spatio-temporal contexts are typically very inefficient. Alternatively, we can slightly modify the classic logistic form, in the following way:

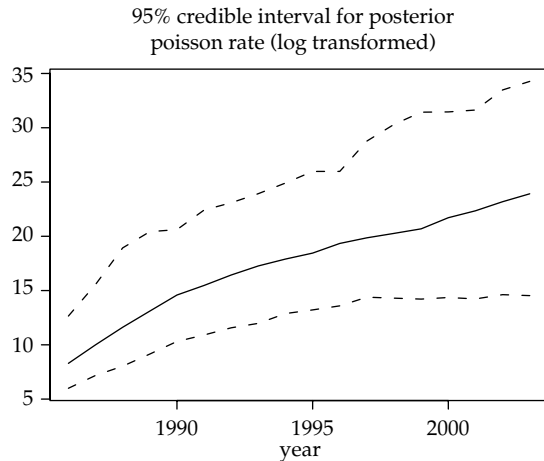


Figure 8.6. Credible interval for the posterior distribution of the log Poisson rate (i.e. $\log(\lambda_t)$) averaged over space for years 1986–2003.

$$\begin{aligned} \mathbf{u}_t = & \mathbf{H}(\delta, \Delta_t, \Delta_x, \Delta_y)\mathbf{u}_{t-\Delta_t} \\ & + \mathbf{H}_B(\delta, \Delta_t, \Delta_x, \Delta_y)\mathbf{u}_{t-1}^B + \beta_0\mathbf{u}_{t-1} \\ & - \beta_1 \text{diag}(\mathbf{u}_{t-1})\mathbf{u}_{t-2} + \eta_t, \end{aligned} \tag{8.27}$$

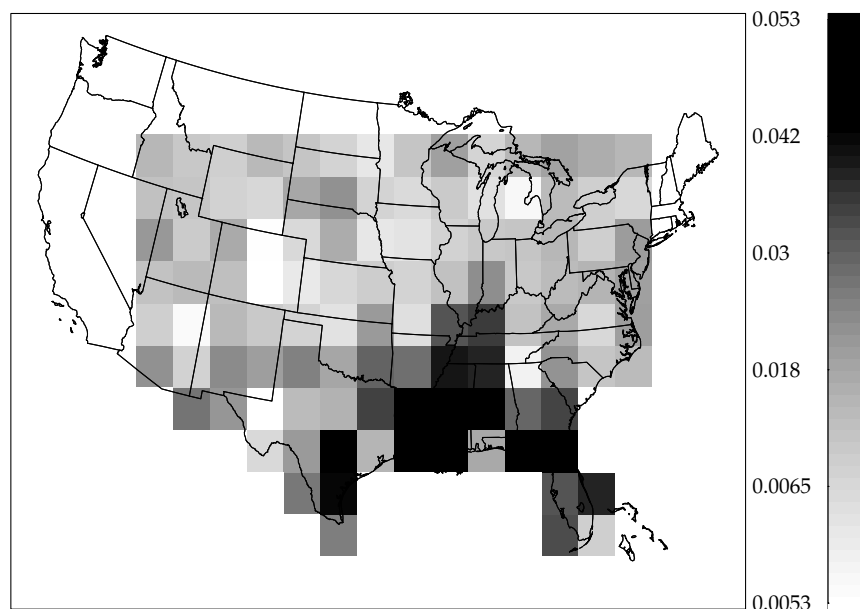


Figure 8.7. Posterior mean of δ , the diffusion coefficients.

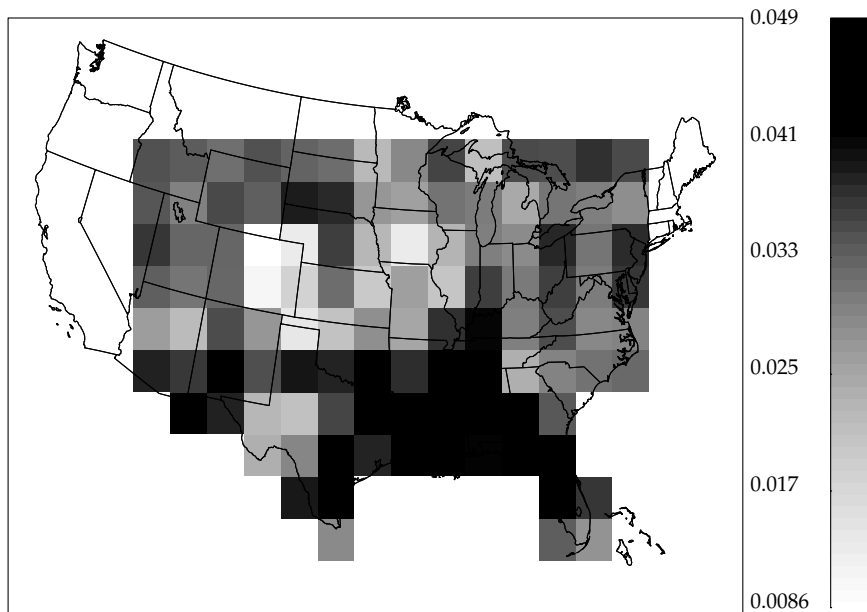


Figure 8.8. Posterior standard deviation of δ , the diffusion coefficients.

where we have set $\beta_0 = \gamma_0$ and $\beta_1 = \gamma_0\gamma_1$ from (8.18). More importantly, we have replaced \mathbf{u}_{t-1} in the last nonnoise term on the right hand side with \mathbf{u}_{t-2} . This simple modification is in the spirit of the original reaction–diffusion model, but the \mathbf{u}_{t-2} term makes it possible to derive analytically the full-conditional for the \mathbf{u}_t 's, potentially improving computational efficiency. This model is investigated in Hooten and Wikle (2005), and it appears to fit the data better than the model presented here. One could check this formally by considering Bayesian model selection.

8.6 Summary and conclusion

This chapter is meant to be a case study of how one can include PDE-based priors for ecological processes in a hierarchical Bayesian spatio–temporal dynamic model. We discussed statistical spatio–temporal dynamical models and mentioned that the critical modeling and implementation issues are related to efficient parameterization of the dynamical propagator (or redistribution) matrix. Such parameterizations can be motivated by the redistribution kernels in the theory of IDEs. In

addition, discretized PDE models can be used to parameterize these dynamics. This was the focus of the present case study.

The case study considered the recent invasion of North America by the Eurasian Collared-Dove. In the process stage of the hierarchical model, we used a discretized version of a simple diffusion PDE with spatially varying diffusion coefficients to parameterize the dynamical propagator matrix. The results show that this model does a reasonable job of representing the data, yet suggests that a more representative model might include a mechanism for population growth.

Much work could be done with the case study presented here in terms of model selection and evaluation. However, the current version serves as a fairly complete illustration of how one can implement these models with “real-world” data sets.

Appendix A: MCMC for Eurasian Collared-Dove case study

Recall from the discussion above that our Bayesian hierarchical model for the Eurasian Collared-Dove

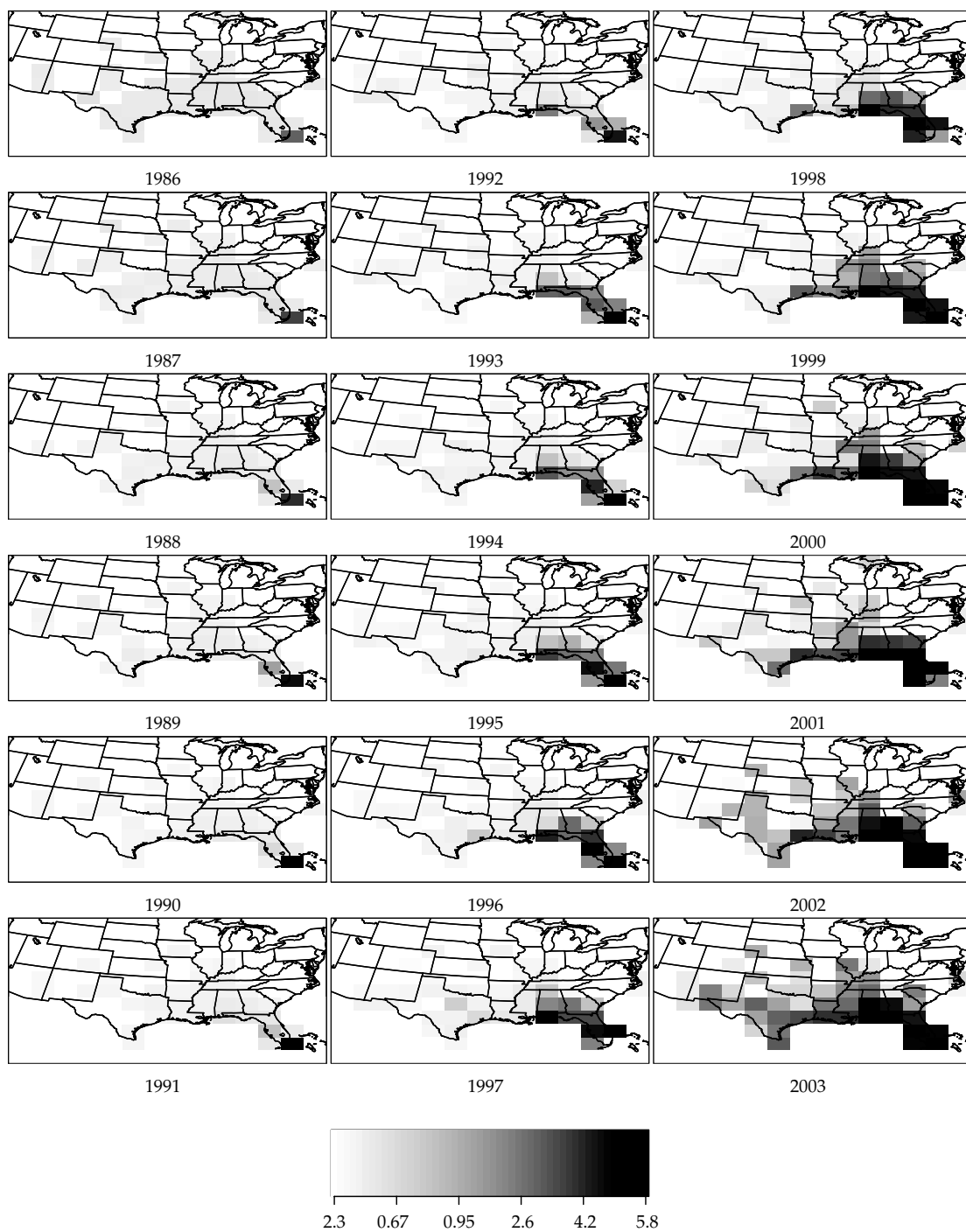


Figure 8.9. Posterior mean of $\log(N(s_i; t)\lambda(s_i; t))$ for years 1986–2003.

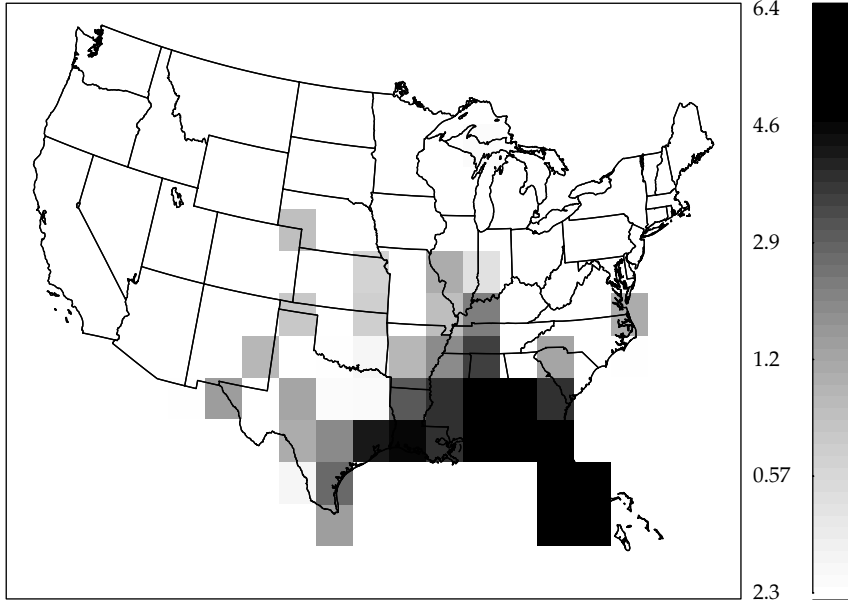


Figure 8.10. Posterior mean of prediction of $\log(N(s_i; t)\lambda(s_i; t))$ for 2004.

data is given as follows:

$$\mathbf{Z}_t | \lambda_t \sim \text{Poi}(\text{diag}(\mathbf{N}_t) \lambda_t), \quad t = 1, \dots, T, \quad (\text{A.1})$$

$$\mathbf{v}_t \equiv \log(\lambda_t) | \mathbf{u}_t, \sigma_\epsilon^2 \sim N(\mathbf{u}_t, \sigma_\epsilon^2 \mathbf{I}), \quad t = 1, \dots, T, \quad (\text{A.2})$$

$$\mathbf{u}_t | \mathbf{u}_{t-1}, \delta, \sigma_\eta^2 \sim N(\mathbf{H}(\delta) \mathbf{u}_{t-1}, \sigma_\eta^2 \mathbf{I}), \quad t = 1, \dots, T, \quad (\text{A.3})$$

$$\mathbf{u}_0 \sim N(\tilde{\mathbf{u}}_0, \Sigma_0), \quad (\text{A.4})$$

$$\delta | \alpha, \sigma_\delta^2 \sim N(\Phi \alpha, \sigma_\delta^2 \mathbf{I}), \quad (\text{A.5})$$

$$\alpha \sim N(\alpha_0, \sigma_\alpha^2 \mathbf{R}_\alpha) \quad (\text{A.6})$$

and

$$\begin{aligned} \sigma_\epsilon^2 &\sim \text{IG}(q_\epsilon, r_\epsilon), & \sigma_\eta^2 &\sim \text{IG}(q_\eta, r_\eta), \\ \sigma_\delta^2 &\sim \text{IG}(q_\delta, r_\delta), & \sigma_\alpha^2 &\sim \text{IG}(q_\alpha, r_\alpha). \end{aligned} \quad (\text{A.7})$$

The Bayesian formulation of the hierarchical model is summarized by the following posterior

distribution:

$$\begin{aligned} &[\lambda_1, \dots, \lambda_T, \mathbf{u}_0, \dots, \\ &\quad \mathbf{u}_T, \delta, \alpha, \sigma_\epsilon^2, \sigma_\eta^2, \sigma_\delta^2, \sigma_\alpha^2 | \mathbf{Z}_1, \dots, \mathbf{Z}_T] \\ &\propto \left\{ \prod_{t=1}^T [\mathbf{Z}_t | \lambda_t][\lambda_t | \mathbf{u}_t, \sigma_\epsilon^2] \right\} \\ &\times \left\{ \prod_{t=1}^T [\mathbf{u}_t | \delta, \mathbf{u}_{t-1}, \sigma_\eta^2][\mathbf{u}_0] \right\} \\ &\times [\delta | \alpha, \sigma_\delta^2][\alpha][\sigma_\epsilon^2][\sigma_\eta^2][\sigma_\delta^2][\sigma_\alpha^2]. \end{aligned} \quad (\text{A.8})$$

There is no analytical representation of this posterior. However, we can use MCMC methods to obtain samples from this posterior distribution. For an overview of MCMC methodologies see Gilks et al. (1996) and Robert and Casella (1999). For complicated spatio-temporal applications of these methods, see Wikle et al. (1998), Berliner et al. (2000), Wikle et al. (2001). For a spatio-temporal diffusion-equation example applied to BBS data see Wikle (2003b).

Below, we present the full-conditional distributions required for the Gibbs sampler MCMC algorithm. In addition, an outline of the sampling

program is presented, followed by the associated R-code.

Full-Conditional Distributions

Based on the hierarchical model described above for the Eurasian Collared-Dove relative abundance through time, the Gibbs sampler cycles through the following full conditional distributions. Specifically, one samples the j -th iteration from the following distributions. Note that we use the notation $[a|\cdot]$ for the full-conditional distribution of the random variable a , where the “dot” to the right of the condition symbol represents all other parameters and the data.

- $[\log(\lambda_t(\mathbf{s}_i))|\cdot]$. For notational convenience, let $v_{it} = \log(\lambda_t(\mathbf{s}_i))$. For $t = 1, \dots, T$ and $i = 1, \dots, n$ we sample from this full-conditional by utilizing the Metropolis-Hastings (e.g. see Robert and Casella 1999) procedure:

1. Generate $v_{it}^* \sim N(v_{it}^{(j-1)}, \zeta)$ and compute the ratio:

$$r = \frac{[Z_t(\mathbf{s}_i)|v_{it}^*][v_{it}^*|\mathbf{u}_t^{(j-1)}, \sigma_\epsilon^{2(j-1)}]}{[Z_t(\mathbf{s}_i)|v_{it}^{(j-1)}][v_{it}^{(j-1)}|\mathbf{u}_t^{(j-1)}, \sigma_\epsilon^{2(j-1)}]}.$$

2. Set $v_{it}^{(j)} = v_{it}^*$ with probability $\min(r, 1)$; otherwise, set $v_{it}^{(j)} = v_{it}^{(j-1)}$.

The parameter ζ is a *tuning* parameter in the Metropolis-Hastings algorithm. In theory, it does not affect the estimates, only the way in which they are obtained. If ζ is large then the parameter space is explored more rapidly, but more of the draws are rejected. Smaller values of ζ lead to slower exploration of the parameter space, but with a higher acceptance rate. Thus, one has to try different values of ζ to compromise between the acceptance rate and the exploration of the parameter space. We found $\zeta = 0.1$ to be a reasonable value here.

- $[\mathbf{u}_0|\cdot]$. Sample from $\mathbf{u}_0^{(j)} \sim N(\mathbf{Ab}, \mathbf{A})$ where

$$\mathbf{A} = (\mathbf{H}'\mathbf{H}/\sigma_\eta^{2(j-1)} + \Sigma_0^{-1})^{-1},$$

$$\mathbf{b} = \mathbf{H}'^{(j-1)}\mathbf{u}_1^{(j-1)}/\sigma_\eta^{2(j-1)} + \Sigma_0^{-1}\tilde{\mathbf{u}}_0,$$

where we have suppressed the dependence of \mathbf{H} on δ for notational convenience.

- $[\mathbf{u}_t|\cdot]$, for $t = 1, \dots, T-1$. Sample from $\mathbf{u}_t^{(j)} \sim N(\mathbf{Ab}, \mathbf{A})$, where

$$\mathbf{A} = \left(\mathbf{I}/\sigma_\eta^{2(j-1)} + \mathbf{H}'^{(j-1)}\mathbf{H}^{(j-1)}/\sigma_\eta^{2(j-1)} + \mathbf{I}/\sigma_\epsilon^{2(j-1)} \right)^{-1},$$

$$\mathbf{b} = \mathbf{H}^{(j-1)}\mathbf{u}_{t-1}^{(j)}/\sigma_\eta^{2(j-1)} + \mathbf{H}'^{(j-1)}\mathbf{u}_{t+1}^{(j-1)}/\sigma_\eta^{2(j-1)} + \mathbf{v}_t^{(j)}/\sigma_\epsilon^{2(j-1)},$$

where we let \mathbf{v}_t be the $n_t \times 1$ vectorization of v_{it} .

- $[\mathbf{u}_T|\cdot]$. Sample from $\mathbf{u}_T^{(j)} \sim N(\mathbf{Ab}, \mathbf{A})$ where

$$\mathbf{A} = \left(\mathbf{I}/\sigma_\eta^{2(j-1)} + \mathbf{I}/\sigma_\epsilon^{2(j-1)} \right)^{-1},$$

$$\mathbf{b} = \mathbf{H}^{(j-1)}\mathbf{u}_{T-1}^{(j)}/\sigma_\eta^{2(j-1)} + \mathbf{v}_T^{(j)}/\sigma_\epsilon^{2(j-1)}.$$

- $[\delta|\cdot]$. To facilitate the presentation of this full conditional, note that we can rewrite (8.23) as:

$$\mathbf{u}_t = \mathbf{G}_{t-1}\delta + \mathbf{u}_{t-1} + \eta_t,$$

where \mathbf{G}_{t-1} is a sparse function of \mathbf{u}_{t-1} . Then, one can sample $\delta^{(j)} \sim N(\mathbf{Ab}, \mathbf{A})$ where

$$\mathbf{A} = \left(\sum_{t=1}^T \mathbf{G}_{t-1}'^{(j)}\mathbf{G}_{t-1}^{(j)}/\sigma_\delta^{2(j-1)} + \mathbf{I}/\sigma_\delta^{2(j-1)} \right)^{-1},$$

$$\mathbf{b} = \sum_{t=1}^T \mathbf{G}_{t-1}'^{(j)}(\mathbf{u}_t^{(j)} - \mathbf{u}_{t-1}^{(j)})/\sigma_\delta^{2(j-1)} + \Phi\alpha^{(j-1)}/\sigma_\delta^{2(j-1)}.$$

- $[\alpha|\cdot]$. Sample $\alpha^{(j)} \sim N(\mathbf{Ab}, \mathbf{A})$, where

$$\mathbf{A} = \left(\mathbf{I}/\sigma_\delta^{2(j-1)} + \mathbf{R}_\alpha^{-1}/\sigma_\alpha^{2(j-1)} \right)^{-1}$$

$$\mathbf{b} = \Phi'\delta^{(j-1)}/\sigma_\delta^{2(j-1)} + \mathbf{R}_\alpha^{-1}\alpha/\sigma_\alpha^{2(j-1)}.$$

- $[\sigma_\epsilon^2|\cdot]$. Sample $\sigma_\epsilon^{2(j)} \sim \text{IG}(q, r)$, where $q = q_\epsilon + nT/2$, where n is the number of spatial locations and

$$r = \left(\frac{1}{r_\epsilon} + 0.5 \sum_{t=1}^T (\mathbf{v}_t^{(j)} - \mathbf{u}_t^{(j)})'(\mathbf{v}_t^{(j)} - \mathbf{u}_t^{(j)}) \right)^{-1}.$$

- $[\sigma_\eta^2|\cdot]$. Sample $\sigma_\eta^{2(j)} \sim \text{IG}(q, r)$, where $q = q_\eta + nT/2$ and

$$r = \left(\frac{1}{r_\eta} + 0.5 \sum_{t=1}^T (\mathbf{u}_t^{(j)} - \mathbf{H}^{(j)}\mathbf{u}_{t-1}^{(j)})'(\mathbf{u}_t^{(j)} - \mathbf{H}^{(j)}\mathbf{u}_{t-1}^{(j)}) \right)^{-1}.$$

- $[\sigma_\delta^2 | \cdot]$. Sample $\sigma_\delta^{2,(j)} \sim \text{IG}(q, r)$, where $q = q_\delta + n/2$, and

$$r = \left(1/r_\delta + 0.5(\delta^{(j)} - \Phi\alpha^{(j)})'(\delta^{(j)} - \Phi\alpha^{(j)})\right)^{-1}.$$

- $[\sigma_\alpha^2 | \cdot]$. Sample $\sigma_\alpha^{2,(j)} \sim \text{IG}(q, r)$, where $q = q_\delta + p/2$ (where p is the length of α) and

$$r = \left(1/r_\alpha + 0.5(\alpha^{(j)} - \alpha_0)'(\alpha^{(j)} - \alpha_0)\right)^{-1}.$$

To perform prediction in space and time, we sample from the following distribution after convergence has been established. We simply sample $\mathbf{u}_{T+1}^{(j)}$ from the prior $[\mathbf{u}_{T+1}^{(j)} | \mathbf{u}_T^{(j)}, \delta^{(j)}, \sigma_\eta^{2,(j)}]$, then sample $\mathbf{v}_{T+1}^{(j)}$ from its prior $[\mathbf{v}_{T+1}^{(j)} | \mathbf{u}_{T+1}^{(j)}, \sigma_\epsilon^{2,(j)}]$, and get $\lambda_{T+1}^{(j)} = \exp(\mathbf{v}_{T+1}^{(j)})$. We can then get a sample from the predictive distribution of $\mathbf{Z}_{T+1}^{(j)}$ by drawing a sample from the data distribution $[\mathbf{Z}_{T+1}^{(j)} | \lambda_{T+1}^{(j)}]$.

Sketch of MCMC Program

The following algorithm could be used to implement the MCMC procedure.

```

*** Choose MCMC parameters
number of iterations
number to burn-in
how often to save matrices and vectors

*** Choose hyperparameters and other
constants
PHI-matrix
finite difference parameters
prior for alpha
inverse gamma parameters (q,r)

*** Choose starting values
v, u, delta, variances

make H matrix

*** Define variables to save samples
for scalars, save all samples
for vectors and matrices,
save every so often
for vectors and matrices,
keep running sum past burn-in
in order to calculate means

```

```

optional: use batching and one
pass calculation of
variance to get estimates of
variability for
matrices and vectors

```

```

*** Main MCMC Loop
for k = 1 to (number of iterations)

```

```

**** sample v(t)
for t = 1 to T
sample v(t) from its full
conditional
set lambda(t) = exp(v(t))
end

```

```

**** sample u(t)
for t = 1 to (T-1)
sample u(t) from its full
conditional
make G(t)
end
sample u(T) from its full
conditional
make G(T)

```

```

**** sample delta
sample delta from its full
conditional
make H

```

```

*** sample alpha
sample alpha from its full
conditional

```

```

*** sample sigma2_epsilon
sample sigma2_epsilon from its full
conditional

```

```

*** sample sigma2_eta
sample sigma2_eta from its full
conditional

```

```

*** sample sigma2_delta
sample sigma2_delta from its full
conditional

```

```

*** sample sigma2_alpha
sample sigma2_alpha from its full
conditional

```

```

%*** Save samples
save all scalar variables
if k > nburn
  update sums for vector and
  matrix variables
  optional: save batching sums
  possibly save matrices and
  vectors if required

  save samples for predictions
  (time T+1)
end
end %main MCMC loop

find means and variances

```

Sample R Code

Note that this sample code is provided as an illustration. It has not been extensively tested and the authors make no claim regarding the accuracy of the code. Note also that this code is “project specific,” meaning that it contains numerous specifications and subroutines that are unique to the data and model considered in this example. The code is given only to illustrate how to employ the above methods, it is not intended (and will not function) for use with other datasets without substantial modification.

```

dgrevised <- function(ngibbs,nburn,
  matsave,lamsave,Z,grdlocs){
#
# (Revised 20050119 Mevin Hooten,
  originally coded 20040528)
# Implements gibbs sampler for
  space-time Bayesian diffusion model
# for Eurasian Collared-Dove data.
# Z is an n x T matrix of the data
#
###
### Data specific variables and
  functions
###
n=111
xp1=c(7:12,14:20,21:27,29:36,38:46,
  48:55,0,0,56:63,64:71,72:79,
  80:87,88:95,98:103,0,0,104:111,
  0,0,rep(0,6))
xm1=c(rep(0,6),1:6,0,7:13,14:20,0,
  21:28,0,29:37,0,38:45,48:55,56:63,
  64:71,72:79,80:87,0,0,88:93,
  96:103)
yp1=c(0,1:5,0,7:12,0,14:19,0,21:27,0,
  29:36,0,38:46,0,48:54,0,56:62,
  0,64:70,0,72:78,0,80:86,0,88:96,
  0,98:102,0,104,0,106:110)
ym1=c(2:6,0,8:13,0,15:20,0,22:28,0,
  30:37,0,39:47,0,49:55,0,57:63,0,
  65:71,0,73:79,0,81:87,0,89:97,0,
  99:103,0,105,0,107:111,0)

XP1 <- function(W){
  XP1out <- matrix(0,n,1)
  XP1out[(1:n)[xp1!=0],]
  <- W[xp1[xp1!=0],]
  XP1out[(1:n)[xp1==0],] <- 0
  XP1out
}
XM1 <- function(W){
  XM1out <- matrix(0,n,1)
  XM1out[(1:n)[xm1!=0],]
  <- W[xm1[xm1!=0],]
  XM1out[(1:n)[xm1==0],] <- 0
  XM1out
}
YP1 <- function(W){
  YP1out <- matrix(0,n,1)
  YP1out[(1:n)[yp1!=0],]
  <- W[yp1[yp1!=0],]
  YP1out[(1:n)[yp1==0],] <- 0
  YP1out
}
YM1 <- function(W){
  YM1out <- matrix(0,n,1)
  YM1out[(1:n)[ym1!=0],]
  <- W[ym1[ym1!=0],]
  YM1out[(1:n)[ym1==0],] <- 0
  YM1out
}

Tb=matrix(0,n,n)
Tc=matrix(0,n,n)
Td=matrix(0,n,n)
Te=matrix(0,n,n)
for(i in 1:n){
  if(xm1[i]!=0){Tb[i,xm1[i]]=1}
  if(xp1[i]!=0){Tc[i,xp1[i]]=1}

```

```

    if (ym1[i] != 0) {Td[i,ym1[i]] = 1}
    if (yp1[i] != 0) {Te[i,yp1[i]] = 1}
  }

makeH <- function(gx,gy,Dvec) {
  a=1-2*gx*Dvec-2*gy*Dvec
  b=(-gx/4)*(XP1(Dvec)-XM1(Dvec))
  +gx*Dvec
  c=(gx/4)*(XP1(Dvec)-XM1(Dvec))
  +gx*Dvec
  d=(-gy/4)*(YP1(Dvec)-YM1(Dvec))
  +gy*Dvec
  e=(gy/4)*(YP1(Dvec)-YM1(Dvec))
  +gy*Dvec
  Fa=(diag(as.vector(a),length(a)))
  Fb=(diag(as.vector(b),
    length(b)))*%Tb
  Fc=(diag(as.vector(c),
    length(c)))*%Tc
  Fd=(diag(as.vector(d),
    length(d)))*%Td
  Fe=(diag(as.vector(e),
    length(e)))*%Te
  H=Fa+Fb+Fc+Fd+Fe
  H
}

makeG <- function(gx,gy,uvec) {
  atilda=(-2*gx-2*gy)*uvec+gx
  *(XP1(uvec)+XM1(uvec))
  +gy*(YM1(uvec)+YP1(uvec))
  btilda=(-gx/4)*(XP1(uvec)-XM1(uvec))
  ctilda=(gx/4)*(XP1(uvec)-XM1(uvec))
  dtilda=(-gy/4)*(YP1(uvec)-YM1(uvec))
  etilda=(gy/4)*(YP1(uvec)-YM1(uvec))
  Fatilda=(diag(as.vector(atilda),
    length(atilda)))
  Fbtilda=(diag(as.vector(btilda),
    length(btilda)))*%Tb
  Fctilda=(diag(as.vector(ctilda),
    length(ctilda)))*%Tc
  Fdtilda=(diag(as.vector(dtilda),
    length(dtilda)))*%Td
  Fetilda=(diag(as.vector(etilda),
    length(etilda)))*%Te
  G=Fatilda+Fbtilda+Fctilda+Fdtilda
  +Fetilda
  G
}

getdist <- function(datalocs) {
  n <- dim(datalocs)[1]
  Cdatloc=datalocs[,1] + complex(1,,1)
  *(datalocs[,2])
  Cgrdloc=datalocs[,1] - complex(1,,1)
  *(datalocs[,2])
  Dst=Mod(Cdatloc%%matrix(1,1,n)
  -Conj(t(Cgrdloc%
  %%matrix(1,1,n))))
  Dst
}

###
### Hyper-parameters and other
### constants
###
Dst=getdist(grdlocs)
expcorr=exp(-4*Dst)
p=1
Phi=eigen(expcorr)$vectors
PHI=Phi
LAMBDA=eigen(expcorr)$values
Phi=Phi[,1:p]
Phi2diag=matrix(diag(t(Phi)*%Phi),
  p,1)
deltat=1
deltax=1
deltay=1
Ralpha=diag(LAMBDA[1:p],p)
Ralphainv=solve(Ralpha)
Ralphainvdiag=matrix(diag(Ralphainv),
  dim(Ralpha)[2],1)
qep=2.8 # mu=2
rep=.2777778 # var=5
qeta=2.9 # mu=3
reta=0.1754386 # var=10
qD=2.8
rD=.2777778
qalpha=2.8
ralpha=.2777778

###
### Initialize Variables
###

saveidx=1
saveidx2=1
m=floor((ngibbs-nburn)/matsave)
l=floor((ngibbs-nburn)/lamsave)
l=l+1
m=m+1

```

```

T=18
vsave=array(0,c(n,T,m))
vsum=matrix(0,n,T)
v=matrix(0,n,T)
usave=array(0,c(n,(T+1),m))
usum=matrix(0,n,(T+1))
u=matrix(0,n,(T+1))
lambdasave=array(0,c(n,T,m))
lambdasum=matrix(0,n,T)
lambda=matrix(0,n,T)
lamsumsave=matrix(0,1,(T+1))
Dsave=matrix(0,n,m)
Dsum=matrix(0,n,1)
D=matrix(0,n,1)
alphasave=matrix(0,p,m)
alphasum=matrix(0,p,1)
alpha=matrix(0,p,1)
alpha0=matrix(0,p,1)
gx=deltat/(deltax^2)
gy=deltat/(deltay^2)
sigma2ep=matrix(0,1,ngibbs)
sigma2eta=matrix(0,1,ngibbs)
sigma2D=matrix(0,1,ngibbs)
sigma2alpha=matrix(0,1,ngibbs)
G <- array(0,c(n,n,(T+1)))
upredM <- matrix(0,n,1)
vpredM <- matrix(0,n,1)
lampredM <- matrix(0,n,1)
ZpredM <- matrix(0,n,1)

###
### Starting Values
###

v=log(Z+.1)
Dvec=.4*matrix(1,n,1)
sigma2ep[,1]=2
sigma2eta[,1]=.1
sigma2D[,1]=.1
sigma2alpha[,1]=.1

H=makeH(gx,gy,as.matrix(Dvec))
onesn=matrix(1,n,1)
util0=matrix(0,n,1)
sigma0inv=diag(n)*.1

###
### Main Gibbs Loop
###

for(k in 2:ngibbs){
  cat(k, " ")

  ###
  ### Sample v
  ###

  for(t in 1:T){
    llold <- dpois(Z[,t],N[,t]
      *exp(v[,t]),log=TRUE)
      +log(dnorm(v[,t],u[(1+t)],
        sqrt(sigma2ep[(k-1)])))
    vc <- rnorm(n,v[,t],.5)
    llnew <- dpois(Z[,t],N[,t]
      *exp(vc),log=TRUE)
      +log(dnorm(vc,u[(1+t)],
        sqrt(sigma2ep[(k-1)])))
    r <- runif(n) < exp(llnew-llold)
    v[r,t] <- vc[r]
    lambda[,t] <- exp(v[,t])
  }

  ###
  ### Sample u
  ###

  G[,,1]
  <- makeG(gx,gy,as.matrix(u[,1]))
  HprimeH <- t(H)%*%H
  Hones <- H%*%onesn

  tvar <- solve(HprimeH/
    sigma2eta[(k-1)] + sigma0inv)
  tmn <- tvar%*%t(t(u[,1+1])%*%H/
    sigma2eta[(k-1)]
    + t(util0)%*%sigma0inv)
  u[,1+0] <- tmn + t(chol(tvar))
  %*%matrix(rnorm(n),n,1)

  for(t in 1:(T-1)){
    ucov <- (solve((diag(n)/
      sigma2eta[(k-1)])
      + (HprimeH)/sigma2eta[(k-1)]+
      (diag(n)/sigma2ep[(k-1)])))
    umn <- ucov%*%t(t(H%*%u[(1+t-1)])/
      sigma2eta[(k-1)] +
      t(u[(1+t+1)])%*%H/sigma2eta[(k-1)]
      + t(v[,t])/sigma2ep[(k-1)])
    u[(1+t)] <- umn + t(chol(ucov))
    %*%matrix(rnorm(n),n,1)
    G[,,(1+t)] <- makeG
      (gx,gy,as.matrix(u[(1+t)]))
  }
}

```

```

uTcov <- solve((diag(n)/sigma2eta
[, (k-1)])+(diag(n)/
sigma2ep[, (k-1)]))
uTmn <- uTcov%%t(t(H%%u[, (1+T-1)])/
sigma2eta[, (k-1)] +
t(v[,t])/sigma2ep[, (k-1)])
u[, (1+T)] <- uTmn + t(chol(uTcov))
%%matrix(rnorm(n),n,1)
G[, (1+T)] <- makeG
(gx,gy,as.matrix(u[, (1+T)]))

###
### Sample D
###

Gsum <- 0
usumtmp <- 0
for(t in 1:T){
  Gsum <- Gsum + t(G[, (1+t-1)])
  %%G[, (1+t-1)]
  usumtmp <- usumtmp + t(u[, (1+t)]
  -u[, (1+t-1)])%%G[, (1+t-1)]
}
Dcov <- solve((diag(n)/
sigma2D[, (k-1)])
+(Gsum/sigma2eta[, (k-1)]))
Dmn <- Dcov %% (Phi%%alpha)/
sigma2D[, (k-1)]+
t(usumtmp)/sigma2eta[, (k-1)])
D <- Dmn + t(chol(Dcov))
%%matrix(rnorm(n),n,1)
D <- matrix(D,n,1)
H <- makeH(gx,gy,matrix(D,n,1))

###
### Sample alpha
###

littlem <- Phi2diag/sigma2D[, (k-1)]
+ Ralphainvdiag/
sigma2alpha[, (k-1)]
piece2 <- t(t(D)%%Phi/
sigma2alpha[, (k-1)]+
t(alpha0)%%Ralphainv/
sigma2alpha[, (k-1)])
alpha <- piece2/littlem
+ (littlem^(-.5))
*matrix(rnorm(p),p,1)

###
### Sample sigma2ep
###

vusumtmp <- 0
for(t in 1:T){vusumtmp <- vusumtmp
+ t(v[,t]-u[, (1+t)])
%%(v[,t]-u[, (1+t)])}
sigma2ep[,k] <- rgamma(1,qep
+ n*T/2,, ((1/rep)
+.5*vusumtmp)^(-1))^(-1)

###
### Sample sigma2eta
###

umusumtmp <- 0
for(t in 1:T){umusumtmp
<- umusumtmp + t(u[, (1+t)]-
(H%%(u[, (1+t-1)]))%%(u[, (1+t)]
-(H%%(u[, (1+t-1)])))}
sigma2eta[,k] <- rgamma(1,qeta
+ n*T/2,, ((1/reta)+
.5*umusumtmp)^(-1))^(-1)

###
### Sample sigma2D
###

sigma2D[,k] <- rgamma(1,qD
+ n/2,, ((1/rD)
+.5*(t(D-Phi%%alpha)%%
(D-Phi%%alpha)))^(-1))^(-1)

###
### Sample sigma2alpha
###

sigma2alpha[,k] <- rgamma(1,qalpha
+ p/2,, ((1/ralpha)+
.5*(t(alpha-alpha0)%%Ralphainv
%%(alpha-alpha0)))^(-1))^(-1)

###
### updating and saving variables
###
if(k > nburn){
  vsum <- vsum + v
  usum <- usum + u
  lambdasum <- lambdasum + lambda
  Dsum <- Dsum + D
  alphasum <- alphasum + alpha

```



```

###
### Predictions
###

upred <- H%*%u[,1+T]
  + sqrt(sigma2eta[,k])
  *matrix(rnorm(n),n,1)
upredM <- upredM + upred
vpred <- upred
  + sqrt(sigma2ep[,k])
  *matrix(rnorm(n),n,1)
vpredM <- vpredM + vpred
lampredM <- lampredM
  + exp(vpred)
ZpredM <- ZpredM
  + matrix(rpois(n,exp(vpred)),n,1)
if(k%lamsave==0){
  lamsumsave[saveidx2,]
  <- apply(cbind(lambda,
    exp(vpred)),2,sum)
  saveidx2 <- saveidx2 + 1
}
if(k%matssave==0){
  vsave[, ,saveidx] <- as.matrix(v)
  usave[, ,saveidx] <- as.matrix(u)
  lambdasave[, ,saveidx]
  <- as.matrix(lambda)
  Dsave[,saveidx] <- D
  alphasave[,saveidx] <- alpha
  saveidx <- saveidx + 1
}
}
} # end main gibbs loop
cat(" n")

###
### Calculating means from sums
###

vmn <- vsum/(ngibbs-nburn)
lambdamn <- lambdasum/(ngibbs-nburn)
umn <- usum/(ngibbs-nburn)
Dmn <- Dsum/(ngibbs-nburn)
alphamn <- alphasum/(ngibbs-nburn)

upredM <- upredM/(ngibbs-nburn)
vpredM <- vpredM/(ngibbs-nburn)
lampredM <- lampredM/(ngibbs-nburn)
ZpredM <- ZpredM/(ngibbs-nburn)

list(vsave=vsave,usave=usave,
  lambdasave=lambdasave,
  upredM=upredM,
  vpredM=vpredM,lampredM=lampredM,
  ZpredM=ZpredM,Dsave=Dsave,
  alphasave=alphasave,vmn=vmn,
  lambdamn=lambdamn,umn=umn,
  Dmn=Dmn,alphamn=alphamn,
  sigma2ep=sigma2ep,
  lamsumsave=lamsumsave,
  sigma2eta=sigma2eta,
  sigma2D=sigma2D,
  sigma2alpha=sigma2alpha,
  PHI=PHI,LAMBDA=LAMBDA)
}

```