8

Dynamic Bayesian Models for Discrete-Valued Time Series

Dani Gamerman, Carlos A. Abanto-Valle, Ralph S. Silva, and Thiago G. Martins

CONTENTS

8.1 Introduction ........................................................... 165
8.2 MCMC ........................................................................ 168
  8.2.1 Updating the States ............................................ 168
    8.2.1.1 Single-Move Update for the States .............. 169
    8.2.1.2 Multimove Update for the States ............... 169
8.3 Sequential Monte Carlo ............................................. 170
  8.3.1 Particle Filter ..................................................... 170
  8.3.2 Adaptive Random Walk Metropolis Sampling ...... 171
8.4 INLA ........................................................................ 172
  8.4.1 INLA Methodology ............................................. 173
  8.4.2 R-INLA through Examples ................................ 174
8.5 Applications ............................................................ 175
  8.5.1 Deep Brain Stimulation ........................................ 175
    8.5.1.1 Computation Details ................................. 175
    8.5.1.2 Results ..................................................... 176
  8.5.2 Poliomyelitis in the U.S. ....................................... 177
    8.5.2.1 Computation Details ................................. 180
    8.5.2.2 Results ..................................................... 180
8.6 Final Remarks .......................................................... 181
8A Appendix ................................................................... 182
  8A.1 Deep Brain Stimulation ........................................ 182
  8A.2 Poliomyelitis in the U.S. ....................................... 183
References ..................................................................... 184

8.1 Introduction

State-space models (SSMs) have been discussed in the literature for a number of decades. They are models that rely on a decomposition that separates the observational errors from the temporal evolution. The former usually consists of temporally independent specifications that handle the characteristics of the observational process. The latter is devised to describe the temporal dependence at a latent, unobserved level through evolution disturbances. In the most general form, the observational and evolution disturbances may be related, but in a typical set-up they are independent. SSMs were originally introduced for

© 2016 by Taylor & Francis Group, LLC
Gaussian, hence continuous, time series data, but the above decomposition made it easy to extend them to discrete-valued time series. This chapter describes SSMs with a view towards their use for such data.

The use of SSM by the statistical time series community has become widespread since the books of Harvey (1989) and West and Harrison (1997). These books provided an extensive account of the possibilities of SSM from the classical and Bayesian perspectives, respectively. Another surge of interest has occurred more recently with the development of sequential Monte Carlo (SMC) methods, allowing for approximate online inference; see the seminal paper by Gordon et al. (1993).

The basic framework upon which this chapter lies is called the dynamic generalized linear model (DGLM). It is a special case of SSM, and was introduced by West et al. (1985). Consider a discrete-valued time series $y_1, \ldots, y_T$ and let $EF(\mu, \phi)$ denote an exponential family distribution with mean $\mu$ and variance $\phi c(\mu)$, for some mean function $c$. The SSM decomposition of DGLM is given, for $t = 1, \ldots, T$, by the equations

**Observation equation:**

$$y_t \mid x_t, \theta \sim EF(\mu_t, \phi),$$  

**Link function:**

$$g(\mu_t) = z'_t x_t,$$

**System equation:**

$$x_t = G_t x_{t-1} + w_t,$$

where $w_t \mid \theta \sim N(0, W)$, (8.3)

where $z_t$ is a known vector (possibly including covariates) at time $t$, $x_t$ is a time-dependent latent state at time $t$, and $\theta$ is a vector of hyperparameters including $\phi$ and unknown components of $G_t$ and $W$. The model is completed with a prior specification for the initial latent state $x_0$. A Bayesian formulation would also require a prior distribution for the hyperparameter $\theta$. The above model formulation considers only linear models both at the link relation and the system evolution levels. A non-Gaussian evolution with nice integration properties was proposed by Gamerman et al. (2013) to replace (8.3). It includes a few discrete observational models but is not as general as the above formulation.

Usual features of time series can be represented in the above formulation. For example, local linear trends are specified with $z_t = (1, 0)'$, $G_t = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $x_t = (\alpha_t, \beta_t)$. In this case, $\alpha$ represents the local level of the series and $\beta$ represents the local growth in the series. Another common feature of time series is seasonality. There are a few related ways to represent seasonal patterns in time series. Perhaps the simplest representation is the structural form of Harvey (1989) where the seasonal effects $s_t$ are stochastically related via

$$s_t = -(s_{t-1} + s_{t-2} + \cdots + s_{t-p+1}) + \eta_t, \quad \forall t,$$

for seasonal cycles of length $p$. Deterministic or static seasonal terms are obtained in the limiting case of $\eta_t = 0$, a.s., thus implying that $\sum_{i=1}^{p} s_{t-i} = 0$, for all $p$. Evolution 8.3 is recovered by forming the latent component $x_t = (s_t, s_{t-1}, \cdots, s_{t-p+1})'$ with $z_t = (1, 0_{p-1})'$, $G_t = \begin{pmatrix} 1'_{p-1} & 0 \\ I_{p-1} \end{pmatrix}$ and $w_t = (\eta_t, 0_{p-1})$, where $I_m$, $1_m$, and $0_m$ are the identity matrix, vector of 1s and vector of 0s of order $m$, respectively.
By far, the most common discrete-valued specifications are the Poisson and binomial distributions. The Poisson distribution is usually assumed in the analysis of time series of counts. The most popular model for time series of counts is the log-linear dynamic model given by

Observation equation: $y_t | x_t \sim \text{Poisson}(\mu_t)$, for $t = 1, \ldots, T$ (8.5)

Link function: $\log(\mu_t) = z_t'x_t$, for $t = 1, \ldots, T$, (8.6)

with system equation (8.3). For binomial-type data, the most popular model is the dynamic logistic regression given by

Observation equation: $y_t | x_t, \theta \sim \text{Bin}(n_t, \pi_t)$, for $t = 1, \ldots, T$ (8.7)

Link function: $\text{logit}(\pi_t) = z_t'x_t$, for $t = 1, \ldots, T$, (8.8)

with system equation (8.3). Similar models are obtained if the logit link is replaced by the probit or complementary log–log links.

A number of extensions/variations can be contemplated:

- Nonlinear models can be considered at the link relation (8.2) and/or at the system evolution (8.3);
- Some components of the latent state $x_t$ may be fixed over time. The generalized linear models (GLM) (Nelder and Wedderburn, 1972) are obtained in the static, limiting case that all components of $x_t$ are fixed;
- The observational equation (8.1) may be robustified to account for overdispersion (Gamerman, 1997);
- The link function (8.2) may be generalized to allow for more flexible forms via parametric (Abanto-Valle and Dey, 2014) or nonparametric (Mallick and Gelfand, 1994) mixtures; and
- The system equation disturbances may be generalized by replacement of Gaussianity by robustified forms (Meinhold and Singpurwalla, 1989) or by skew forms (Valdebenito et al., 2015).

Data overdispersion is frequently encountered in discrete-valued time series observed in human-related studies. It can be accommodated in the DGLM formulation (8.1) through (8.3) via additional random components in the link functions (8.6) and (8.8). These additional random terms cause extra variability at the observational level, forcing a data dispersion larger than that prescribed by the canonical model. These terms may be included in conjugate fashion, thus rendering negative binomial and beta-binomial to replace Poisson and binomial distributions, respectively leading to hierarchical GLM (Lee and Nelder, 1999). Alternatively, random terms may be added to the linear predictors $z_t'x_t$ in the link equations (Ferreira and Gamerman, 2000). The resulting distributions are also overdispersed but no longer available analytically in closed forms. Their main features resemble those of the corresponding negative binomial and beta-binomial distributions, for $N(0, \sigma^2)$ random terms.

Inference can be performed in two different ways: sequentially or in a single block. From a Bayesian perspective, these forms translate into obtaining the sequence of distributions of $[(x_t, \theta) \mid y^t]$, for $t = 1, \ldots, T$ or $[(x_1, \ldots, x_T, \theta) \mid y^T]$, respectively, where
Handbook of Discrete-Valued Time Series

\[ y^t = \{y^0, y_1, \ldots, y_t\} \] and \( y^0 \) represents the initial information. The sequential approach is obtained via iterated use of Bayes’ theorem

\[
p(x_t, \theta | y^t) \propto p(y_t | x_t, \theta) p(x_t | y^{t-1}, \theta)p(\theta | y^{t-1}), \tag{8.9}
\]

where the first term on the right side is given by (8.1). The second term on the right side is obtained iteratively via

\[
p\left(x_t \mid y^{t-1}, \theta\right) = \int p\left(x_t \mid x_{t-1}, y^{t-1}, \theta\right) p\left(x_{t-1} \mid y^{t-1}, \theta\right) dx_{t-1}, \tag{8.10}
\]

where the first term in the integrand is given by (8.3).

Single block inference is performed by a single pass of Bayes’ theorem as

\[
p\left(x_0, x_1, \ldots, x_T, \theta \mid y_T\right) \propto \prod_{i=1}^{T} p(y_t | x_t, \theta) \times \prod_{i=1}^{T} p(x_t | x_{t-1}, \theta) \times p(x_0) p(\theta), \tag{8.11}
\]

where the terms in the products above are respectively given by (8.1) and (8.3).

Inference may also be performed from a classical perspective. In this case, a likelihood approach would use the above posterior distribution as a penalized likelihood, probably with removal of the prior \( p(\theta) \). This route is pursued in Durbin and Koopman (2001) (see also Davis and Dunsmuir [2015; Chapter 6 in this volume]). This chapter will concentrate on the Bayesian paradigm.

The resulting distributions are untractable analytically for all cases discussed above and approximations must be used. The following sections describe some of the techniques that are currently being used to approximate the required distributions. They are Markov chain Monte Carlo (MCMC), SMC (or particle filters), and integrated nested Laplace approximations (INLA). These techniques do not exhaust the range of possibilities for approximations, but are the most widely used techniques nowadays. Finally, the techniques are applied to real datasets under a variety of model formulations in order to illustrate the usefulness of the SSM formulation.

8.2 MCMC

We describe MCMC methods in the context of the general SSM given in (8.1)–(8.3). A detailed review on this subject can be found in Fearnhead (2011) and Migon et al. (2005). From the Bayesian perspective, inference in a general SSM targets the joint posterior distribution of parameters and hidden states, \( p(\theta, x^T \mid y^T) \), which is given by (8.11). A Markov chain whose state is \( (\theta, x^T) \) and whose stationary distribution is the joint posterior distribution \( p(\theta, x^T \mid y^T) \) is subsequent. A realization of this chain is generated until convergence is reached. After convergence, the following iterations of the chain can be used to form a sample from the posterior distribution. The Gibbs sampler, iteratively drawing samples from \( p(x^T \mid y^T, \theta) \) and \( p(\theta \mid x^T, y^T) \) is the most popular method to sample from such a posterior distribution. In practice, sampling from \( p(\theta \mid x^T, y^T) \) is often easy, whereas designing a sampler for \( p(x^T \mid y^T, \theta) \) is trickier due to the high posterior correlation that usually occurs between the states. Next, we will describe approaches to sample from states.
8.2.1 Updating the States

The simplest procedure is to update the components of the states \( x^T \) one at a time in a single-move fashion (Carlin et al., 1992; Geweke and Tanizaki, 2001). However, due to the severe correlation between states, such a sampler may lead to slow mixing. In such cases it is better to update the states in a multimove fashion as blocks of states \( x^{r,s} = (x_r, x_{r+1}, \ldots, x_s)' \), or update the whole state process \( x^T \) (Shephard and Pitt, 1997; Carter and Kohn, 1994, 1996).

8.2.1.1 Single-Move Update for the States

Carlin et al. (1992) and Geweke and Tanizaki (2001) introduced the Gibbs sampler and the Metropolis–Hastings algorithms to perform inference for nonnormal and nonlinear SSM in a single-move fashion. For sampling states, a sequential sampler that updates each state conditioning on the rest of the states is used. Such an approach is easy to implement and due to the Markovian evolution of the states, the conditional distribution of each state given all the others reduces to conditioning only on its two adjacent states:

\[
p(x_t | y_t, x_{t-1}, x_{t+1}, \theta) \propto \begin{cases} p(y_t | x_t, \theta)p(x_{t+1} | x_t, \theta)p(x_t | x_{t-1}, \theta) & t < T \\ p(y_T | \theta, x_T)p(x_T | x_{T-1}, \theta) & t = T \ (end \ point) \end{cases} \tag{8.12}
\]

In some situations, we can simulate directly from the full conditional distribution, and such moves will always be accepted. Where this is not possible, a Metropolis–Hastings step within the Gibbs sampler can often be implemented. Geweke and Tanizaki (2001) give a detailed discussion of several proposals in this situation.

8.2.1.2 Multimove Update for the States

While single-move samplers are easy to implement, the resulting MCMC algorithms can mix slowly if there is strong dependence in the state process. Updating the states in a multimove fashion could be an alternative approach to overcome this problem. Ideally, we would update the whole state process in one move. The simulation smoother of de Jong and Shephard (1995) can be used to sample the states.

In situations where it is not possible to update the whole state process, Shephard and Pitt (1997) and Watanabe (2004) propose sampling random blocks of the disturbances \( w^{r+1,s} = (w_{r+1}, \ldots, w_s)' \) (equivalently \( x^{r+1,s} \) using as proposal transition density a second order Taylor expansion of \( I_t = \log p(y_t | \hat{\zeta}_t) \) in the full conditional of \( w^{r+1,s} \), where \( h(\hat{\zeta}_t) = d_t = z'_t x_t \). The proposal density is then the multivariate normal with pseudo-observations \( \hat{y}_t = z'_t x_t + \hat{V}_t l'_t(\hat{\zeta}_t) \) and \( V_t = - l''(\hat{\zeta}_t) \), for \( t = r + 1, \ldots, s - 1 \) and \( t = T \). For \( s < T \), we have \( \hat{y}_t = \hat{V}_t \left[ z_t | l'(\hat{\zeta}_t) - l''(\hat{\zeta}_t)z_t x_t \right] + G'_{t+1} W_{t+1}^{-1} x_{t+1} \) and \( V_t = \left[ G'_{t+1} W_{t+1}^{-1} G_{t+1} - l''(\hat{\zeta}_t)z_t z'_t \right]^{-1} \). Then the linear SSM with pseudo-observations \( \hat{y}_t \) is defined as

\[
\hat{y}_t = \begin{cases} z'_t x_t + v_t & t = r + 1, \ldots, s - 1, \text{ and } t = T \\ x_t + v_t & t = s < T \end{cases}
\]

\[
x_t = G_t x_t + w_t, \forall \ t, \tag{8.13}
\]
where $\nu_t$ and $w_t$ are all independent and $\nu_t \sim \mathcal{N}(0, \hat{V}_t)$, $w_t \sim \mathcal{N}(0, W_t)$. Notice that sampling from this distribution $(g)$ is the same as sampling $w_{r+1,s}$ given $x_r, x_{s+1}$ and $\hat{y}_{r+1}, \ldots, \hat{y}_s$ in the above model, which is possible by using the de Jong and Shephard (1995) simulation smoother. Since the distribution $f$ of the disturbances is not bounded by $g$, the Metropolis–Hastings acceptance–rejection algorithm samples from $f$ as recommended by Chib and Greenberg (1995). The expansion blocks $\hat{w}_{r+1,s}$ ($x_{r+1,s}$) are selected as follows. Once an initial expansion block is selected, the auxiliary observations $\hat{y}_t$ are calculated. Next, application of the Kalman filter and a disturbance smoother to the linear Gaussian SSM with the artificial $\hat{y}_t$ yields the mean of $x_{r+1,s}$ conditional on $\hat{x}_{r+1,s}$. By repeating the procedure until the smoothed estimates converge, we obtain the posterior mode $x_{r+1,s}$. According to Shephard and Pitt (1997), the blocks are selected randomly.

Gamerman (1998) suggests the use of a proposal transition density very similar to that of Shephard and Pitt (1997) based on a reparametrization of the model in terms of the system disturbances and sampling from these distributions. The proposal density is the full conditional distribution of $w_t$ in the model with the modified observational equation in (8.13). The reparametrization rewrites the link function in terms of the system disturbances as $g(\mu_t) = \alpha_t = z_t' \sum_{j=1}^{t} G_{t-j} w_j$ with $w_t \sim \mathcal{N}(0, W_t)$, $t = 2, \ldots, T$ and $w_1 \sim \mathcal{N}(a_1, R_1)$, if $G_t = G, \forall t$.

### 8.3 Sequential Monte Carlo

Since the seminal work by Gordon et al. (1993), the SMC methods—also known as particle filter algorithms—have gained popularity as generalization of the importance sampling algorithm. The auxiliary particle filter (Pitt and Shephard, 1999) is another generalization of SMC. Since SMC methods are employed for filtering and smoothing given the parameters, the parameters should first be estimated before estimating the state vector. Recently, Andrieu et al. (2010) proposed the particle MCMC algorithm, which combines two algorithms: Metropolis–Hastings and SMC methods. They showed that if the likelihood is unbiasedly estimated by SMC and is plugged into a Metropolis–Hastings algorithm, then the parameters and states can be sampled from the correct posterior distribution. Nonetheless, designing a Metropolis–Hastings algorithm and tuning it can be cumbersome, especially for some state space models. It is possible to use adaptive Metropolis–Hastings sampling schemes as in Pitt et al. (2012) to overcome this problem. The first scheme is the adaptive random walk Metropolis sampling proposed by Roberts and Rosenthal (2009) (see references therein), and the second one is the adaptive independent Metropolis–Hastings sampling algorithm proposed by Giordani and Kohn (2010).

We show below the combination of the particle filter proposed by Gordon et al. (1993) and the adaptive random walk Metropolis sampling proposed by Roberts and Rosenthal (2009), which enables us to draw from the exact posterior distribution of the parameter vector, including the states. We should keep in mind that there are other more efficient combinations of particle filtering and adaptive sampling methods.

#### 8.3.1 Particle Filter

We describe the particle filter by Gordon et al. (1993). Suppose that we have samples $x_{t-1}^k \sim p(x_{t-1}|y^{t-1}, \Theta)$ for $k = 1, \ldots, K$. First, we take the sample $\hat{x}_t^k \sim p(x_t|x_{t-1}^k)$, for
We can compute the corresponding weights and probabilities by
\[
\delta_i^k = p \left( y_i | \tilde{x}_i^k, \Theta \right) \quad \text{and} \quad \omega_i^k = \frac{\delta_i^k}{\sum_{j=1}^K \delta_j^k}.
\]

The filtering density \( p(x_t | y_1, \ldots, y^t, \Theta) \) can be approximated by \( \left\{ \left( \tilde{x}_i^k, \omega_i^k \right) \right\}_{k=1}^K \), that is,
\[
\hat{p}(x_t | y^t, \Theta) = \sum_{k=1}^K \omega_i^k \Delta \left( x_t - \tilde{x}_i^k \right),
\]
where \( \Delta \) is the Dirac function.

The next step is to resample from this mass function to obtain an equally weighted sample, which we call \( \tilde{x}_i^k \) for \( k = 1, \ldots, K \). We can use the multinomial sampling in this resampling step although stratified sampling reduces the variance of the simulated likelihood. The true likelihood is in fact estimated by the simulated likelihood that is given by
\[
\hat{p}(y_t | y_1, \ldots, y^{t-1}, \Theta) = \frac{1}{K} \sum_{k=1}^K p \left( y_t | \tilde{x}_i^k \right) = \frac{1}{K} \sum_{k=1}^K \delta_i^k.
\]

We now move to the next time step.

### 8.3.2 Adaptive Random Walk Metropolis Sampling

The posterior distribution \( p(\Theta | Y) \) is our target density from which we wish to draw a sample. However, it is computationally difficult to do so directly and we use the Metropolis–Hastings algorithm. Given an initial \( \Theta_0 \), we then generate \( \Theta_j \) for \( j \geq 1 \) from the proposal density \( g_j(\Theta, \Theta^*) \) where \( \Theta^* \) represents previous iteration values of \( \Theta \). Define \( \Theta_j^p \) as the proposed value of \( \Theta_j \), generated from \( g_j(\Theta_j; \Theta_{j-1}) \). We then take \( \Theta_j = \Theta_j^p \) with probability
\[
\alpha(\Theta_{j-1}; \Theta_j^p) = \min \left\{ 1, \frac{p(\Theta_j^p | Y) g_j(\Theta_{j-1}; \Theta_j^p)}{p(\Theta_{j-1} | Y) g_j(\Theta_j^p; \Theta_{j-1})} \right\},
\]
and take \( \Theta_j = \Theta_{j-1} \) otherwise. Under some regularity conditions (Tierney, 1994), the sequence \( \{\Theta_j, j = 1, \ldots, n\} \) converges as \( n \to \infty \) to draws from the target density \( p(\Theta | Y) \).

The adaptive random walk Metropolis proposal of Roberts and Rosenthal (2009) is
\[
g_j(\Theta; \Theta_{j-1}) = \gamma_j \phi_d(\Theta | \Theta_{j-1}, \eta_1 \Omega_1) + (1 - \gamma_j) \Phi_d(\Theta | \Theta_{j-1}, \eta_2 \Omega_2),
\]
where \( d \) is the dimension of \( \Theta \) and \( \phi_d(\cdot | \mu, \Omega) \) is a multivariate \( d \) dimensional normal density with mean \( \mu \) and covariance matrix \( \Omega \). In (8.17), \( \gamma_j = 1 \) for \( j \leq k \), with \( k \geq 1 \) representing the initial iterations, \( \gamma_j = 0.05 \) for \( j > k \); \( \eta_1 = 0.1^2/d \), which makes the sampler to move
locally in small steps; \( \eta_2 = 2.38^2 / d \), which is optimal (Roberts et al., 1997) when the posterior distribution is a multivariate normal; \( \Omega_1 \) is a constant covariance matrix that may be derived from an estimate of the parameters or may simply be the identity matrix; \( \Omega_{2j} \) is the sample covariance matrix of the first \( j - 1 \) iterates (the adaptive step).

On one hand the posterior distribution is considered to be approximated by \( \hat{p}(\Theta|Y) \propto \hat{L}(Y; \Theta)p(\Theta) \). On the other hand, we can take all the random variables used in the particle filter as a vector \( u \) of uniforms and the posterior distribution in an augmented space as \( p(\Theta, u|Y) \propto L(Y; \Theta, u)p(\Theta)p(u) \). We can plug this into a Metropolis–Hastings scheme such as the adaptive random walk Metropolis to draw a sample from \( p(\Theta|Y) \propto L(Y; \Theta)p(\Theta) \).

See Andrieu et al. (2010) and Pitt et al. (2012) for more details. There are efficient ways to filter as a vector side this scope. Looking at (8.1 and 8.2), we see that INLA was originally designed. We refer to Rue et al. (2009) for examples, which are out-friendly interface to the INLA methodology, allowing it to be used routinely, even for those who are not interested in the implementation details behind the program.

As mentioned in Martins et al. (2013), implementation of the INLA methodology requires some expertise in numerical methods and computer programming, in order to achieve efficient computing times. The R (R Core Team, 2013) package INLA, hereafter denoted as R-INLA, was developed to overcome this challenge, and provides a user-friendly interface to the INLA methodology, allowing it to be used routinely, even for those who are not interested in the implementation details behind the program.

The models defined by (8.1)–(8.3) belong to the class of latent Gaussian models for which INLA was originally designed. We refer to Rue et al. (2009) for examples, which are outside this scope. Looking at (8.1 and 8.2), we see that \( \pi(y_i|x_i, \theta_1) = EF(\pi^{-1}(z_i, x_i), \phi) \), \( i = 1, \ldots, T \) with \( \theta_1 = \phi \). From (8.3) we can write

\[
p(x|\theta_2) = p(x_0|\theta_2) \prod_{i=1}^{T} p(x_i|x_{i-1}, \theta_2),
\]

where \( x_0|\theta_2 \sim N(0, W_0) \) and \( x_i|x_{i-1}, \theta_2 \sim N(G_i x_{i-1}, W) \). Since \( x_0|\theta_2 \) and \( x_i|x_{i-1}, \theta_2, i = 1, \ldots, T \) are Gaussian, it can be shown that \( x|\theta_2 \sim N(0, Q^{-1}(\theta_2)) \), with precision matrix given by

\[
Q(\theta_2) = \begin{bmatrix}
W_0^{-1} & G_1^TW_1^{-1} & 0 & \cdots & 0 & 0 \\
W_1^{-1}G_1 & W_1^{-1} + G_2^TW_2^{-1}G_2 & G_2^TW_2^{-1} & \cdots & 0 & 0 \\
0 & W_2^{-1}G_2 & W_1^{-1} + G_3^TW_3^{-1}G_3 & \cdots & 0 & 0 \\
0 & 0 & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & W_1^{-1}G_{T-1} & W_1^{-1} + G_TW_1^{-1}G_T & G_T^TW_T^{-1} \\
0 & 0 & \cdots & 0 & W_1^{-1}G_T & W_1^{-1}
\end{bmatrix}
\]
and $\theta_2$ consists of the unknown parameters within the variance–covariance matrix $W$ and the matrices $G_i, i = 1, \ldots, T$. Therefore, $x|\theta_2$ is a latent Gaussian model with a sparse precision matrix $Q(\theta_2)$, also known as a Gaussian Markov Random Field (GMRF) (Rue and Held, 2005).

Since the dynamic models of interest in this chapter can be written as

$$p(y, x, \theta) = p(y|x, \theta)p(x|\theta)p(\theta),$$

we will describe the INLA methodology in Section 8.4.1, and illustrate how to use the R-INLA package through examples in Section 8.4.2.

### 8.4.1 INLA Methodology

For the hierarchical model described earlier, the joint posterior distribution is given by

$$p(x, \theta|y) \propto p(\theta)p(x|\theta) \prod_{i=1}^{n_d} p(y_i|\eta_i(x), \theta) \propto p(\theta)|Q(\theta)|^{n/2} \exp \left[ -\frac{1}{2} x^T Q(\theta) x + \sum_{i=1}^{n_d} \log \{p(y_i|x_i, \theta)\} \right],$$

and the posterior marginals of interest can be written as

$$p(x_i|y) = \int p(x_i|\theta, y)p(\theta|y)d\theta, \quad i = 1, \ldots, n, \quad (8.18)$$

$$p(\theta_j|y) = \int p(\theta|y)d\theta_{-j}, \quad j = 1, \ldots, m, \quad (8.19)$$

INLA provides approximations $\tilde{p}(\theta|y), \tilde{p}(x_i|\theta, y)$ to $p(\theta|y)$, and $p(x_i|\theta, y)$, plugs them into (8.18) and (8.19), and uses numerical integration to obtain the approximated posterior marginals $\tilde{p}(x_i|y), \tilde{p}(\theta_j|y)$ of interest.

The approximation used for the joint posterior of the hyperparameters $p(\theta|y)$ is

$$\tilde{p}(\theta|y) \propto \frac{p(x, \theta, y)}{p_G(x|\theta, y)|_{x=x}(\theta)},$$

(8.20)
where \( p_G(x|\theta, y) \) is a Gaussian approximation to the full conditional of \( x \), \( p(x|\theta, y) \), obtained by matching the modal configuration and the curvature at the mode, and \( x^* (\theta) \) is the mode of the full conditional for \( x \), for a given \( \theta \). Expression (8.20) is equivalent to the Laplace approximation of a marginal posterior distribution (Tierney and Kadane, 1986), and it is exact if \( p(x|y, \theta) \) is Gaussian.

For \( p(x_i|\theta, y) \), three options are available, and they vary in terms of speed and accuracy. The fastest option, \( p_G(x_i|\theta, y) \), is to use the marginals of the Gaussian approximation \( p_G(x|\theta, y) \), which is already computed when evaluating expression (8.20). The only extra cost in obtaining \( p_G(x_i|\theta, y) \) is to compute the marginal variances from the sparse precision matrix of \( p_G(x|\theta, y) \), see Rue et al. (2009) for details. The Gaussian approximation often gives reasonable results, but it may contain errors in the location and/or errors due to its lack of skewness (Rue and Martino, 2007). The more accurate approach would be to again use a Laplace approximation, denoted by \( p_{LA}(x_i|\theta, y) \), with a form similar to (8.20), that is,

\[
p_{LA}(x_i|\theta, y) \propto \frac{p(x, \theta, y)}{p_G(x_{-i}|x_i, \theta, y)} \bigg|_{x_i = x^* (x, \theta)}, \tag{8.21}
\]

where \( x_{-i} \) represents the vector \( x \) with its \( i \)th element excluded and \( p_G(x_{-i}|x_i, \theta, y) \) is the Gaussian approximation to \( x_{-i}|x_i, \theta, y \) and \( x^*_{-i}(x_i, \theta) \) is the modal configuration. A third option \( p_{SLA}(x_i|\theta, y) \), called simplified Laplace approximation, is obtained by doing a Taylor expansion on the numerator and denominator of (8.21) up to third order, thus correcting the Gaussian approximation for location and skewness with a much lower cost when compared to \( p_{LA}(x_i|\theta, y) \). We refer to Rue et al. (2009) for a detailed description of the Gaussian, Laplace and simplified Laplace, approximations to \( p(x_i|\theta, y) \).

Finally, once we have the approximations \( \tilde{p}(\theta|y) \), \( \tilde{p}(x_i|\theta, y) \) described earlier, the integrals in (8.18) and (8.19) are numerically approximated by discretizing the \( \theta \) space through a grid exploration of \( \tilde{p}(\theta|y) \). Details about this grid exploration can be found in Martins et al. (2013).

### 8.4.2 R-INLA through Examples

The syntax for the \texttt{R-INLA} package is based on the built-in \texttt{glm} function in \texttt{R}, and a basic call starts with

```r
formula = y ~ a + b + a:b + c*d + f(idx1, model1, ...) + f(idx2, model2, ...),
```

where \texttt{formula} describes the structured additive linear predictor \( \eta(x) \). Here, \( y \) is the response variable, the term \( a + b + a:b + c*d \) holds similar meaning as in the builtin \texttt{glm} function in \texttt{R} and is then responsible for the fixed effects specification. The \( f() \) terms specify the general Gaussian random effects components of the model. In this case we say that both \texttt{idx1} and \texttt{idx2} are latent building blocks that are combined together to form a joint latent Gaussian model of interest. Once the linear predictor is specified, a basic call to fit the model with \texttt{R-INLA} takes the following form:

```r
result = inla(formula, data = data.frame(y, a, b, c, d, idx1, idx2), family = "gaussian").
```
After the computations, the list variable `result` will hold an S3 object of class "inla", from which summaries, plots, and posterior marginals can be obtained. We refer to the package website http://www.r-inla.org for more information about model components available to use inside the `f()` functions as well as more advanced arguments to be used within the `inla()` function. In the next section, we show how R-INLA can be used to fit dynamic models. Ruiz-Cárdenas et al. (2011) provide more detailed examples.

8.5 Applications

8.5.1 Deep Brain Stimulation

A binary time series of infant sleep status was recorded in a $T = 120$ min electroencephalographic (EEG) sleep pattern study (Stoffer et al., 1998). Let $y_t$ be the indicator of REM sleep cycle. Two time-varying covariates are considered: $z_{t1}$, the number of body movements during minute $t$; and $z_{t2}$, the number of body movements not due to sucking during minute $t$. As in Czado and Song (2008), our main objective is to investigate whether or not the probability of being in the REM sleep status is significantly related to the two types of body movements, $z_{t1}$ or $z_{t2}$. Our analysis considers the probit link and the Student-t link (shown here in the equations) and assumes that

\[ y_t \sim \text{Ber}(\pi_t) \quad t = 1, \ldots, T, \]

\[ \pi_t = T_\nu(\beta_0 + \beta_1 z_{t1} + \beta_1 z_{t2} + x_t), \]

\[ x_t = \delta x_{t-1} + \tau \eta_t, \]

where the innovations $\eta_t$ are assumed to be mutually independent and normally distributed with mean zero and unit variance and $T_\nu$ is the d.f. of the $t_\nu(0,1)$ distribution. Clearly $x_t$ represents a time-specific effect on the observed process. It follows from the one-to-one relationship (8.23) that $\pi_t = P(y_t = 1 \mid \beta, z_t, x_t) = T_\nu(\beta_0 + \beta_1 z_{t1} + \beta_1 z_{t2} + x_t)$, where $\beta = (\beta_0, \beta_1, \beta_2)'$, $z_t = (1, z_{t1}, z_{t2})'$. We also assume that $| \delta | < 1$, that is, the latent state process is stationary and $x_0 \sim N(0, \kappa)$, where $\kappa = \tau^2/(1 - \delta^2)$. The model is completed with priors $\delta \sim N(-1.1)(0.95, 100)$, $\kappa \sim IG(0.01, 0.01)$, and $\beta \sim N_3(0, \Sigma_0)$, where $\beta_0 = 0$ and $\Sigma_0 = 1000I_3$, $0$ indicates a 3 x 1 vector of zeros and $I_3$ the identity matrix of order 3. For $\nu$, we assume a noninformative prior as in Fonseca et al. (2008).

8.5.1.1 Computation Details

As in Abanto-Valle and Dey (2014), we adopt the so-called threshold approach (Albert and Chib, 1993), where $y_t$ is created through dichotomization of a latent continuous process $Z_t$, given by the one-to-one correspondence

\[ y_t = 1 \Leftrightarrow u_t > 0, \quad t = 1, \ldots, T. \]
With the unobservable or latent threshold variable vector \( u_t = (u_1, \ldots, u_T) \), equations (8.22) and (8.23) can be rewritten as

\[
u_t = z_t' \beta + x_t + \lambda_t^{-1/2} \epsilon_t, \tag{8.26}\]

\[
\lambda_t \sim \mathcal{G}(\nu_2, \nu_2), \tag{8.27}\]

where the innovations \( \epsilon_t \) are assumed to be mutually independent and normally distributed with mean zero and unit variance. A similar data augmentation scheme can be used if the link function is logistic instead of probit (Polson et al., 2013). Equations (8.24) and (8.26), conditioned on \( \delta \), the vector \( \beta \), and the mixing variable \( \lambda_t \), jointly represent a linear state space model, so the algorithm of de Jong and Shephard (1995) is used to simulate the states.

We fit an SSM to the binary observations using probit and Student-t links. For each case, we conducted MCMC simulation for 50,000 iterations. For both cases, the first 10,000 draws were discarded as a burn-in period. In order to reduce the autocorrelation between successive values of the simulated chain, every 20th value of the chain was stored. Posterior means and 95% credible intervals were calculated using the resulting 2000 values.

For SMC, we ran the adaptive random walk Metropolis for 300,000 iterations and discarded the first 100,000 while the particle filter was run with 100 particles only. Note that there are some differences in the estimates, but not much on the persistence parameter \( \phi \).

The R-INLA commands required for the analysis are described in the Appendix.

### 8.5.1.2 Results

The results obtained with the different approximation schemes are summarized later in this chapter. They show reasonable agreement between the different schemes for any given model. Results for the Student-t link are not presented for the INLA analysis because this option is not yet available in R-INLA.

The main results for the static parameters are summarized in Tables 8.1 and 8.2. In both models, the posterior means of \( \delta \) are around 0.93–0.95, showing higher persistence of the autoregressive parameter for states variables and thus in the binary time series. The heaviness of the tails is measured by the shape parameter \( \nu \) in the BSSM-T model. In Table 8.2, the posterior mean of \( \nu \) is around 10–15. This result seems to indicate that the measurement errors of the \( u_t \) threshold variables are better explained by heavy-tailed distributions, as a consequence the \( t \)-links could be more convenient than the probit link. We found empirically that the influence of the number of body movements (\( z_1 \)) is marginal, since the corresponding 95% credible intervals for \( \beta_1 \) contain the zero value. On the other hand, the influence of the number of body movements not due to sucking (\( x_2 \)) is detected to be statistically significant. The negative value of the posterior mean for \( \beta_2 \) shows that a higher number of body movements not due to sucking will reduce the probability of the infant being in REM sleep.

Figures 8.1 and 8.2 show the posterior smoothed means of the probabilities \( \pi_t \) for both links considered. They show substantial agreement between the different schemes for both models. A more thorough comparison is also possible using fit criteria such as BIC or DIC. We found some differences between the fits from the different models, but in general the results are in accordance with Czado and Song (2008). Such difference between the methods
are expected and are mostly within the corresponding Monte Carlo errors associated with the sampling procedures. Error evaluation for the INLA is yet to be derived.

8.5.2 Poliomyelitis in the U.S.

In this section, we consider a time series of monthly counts of cases of poliomyelitis in the United States between January 1970 and December 1983 (Zeger, 1988). The observations are displayed in Figure 8.3. This dataset has been frequently analyzed in the literature,
FIGURE 8.1
Probit link: Posterior smoothed mean of $\pi_t$ applied to the infant sleep status data set.

FIGURE 8.2
Student-t link: Posterior smoothed mean of $\pi_t$ applied to the infant sleep status data set.
for example, by Chan and Ledolter (1995), Le Strat and Carrat (1999), and Davis and Rodriguez-Yam (2005). We concentrate below on the estimation of the mean response and the hyperparameters.

We adopt the loglinear seasonal Poisson SSM defined by

\[
\begin{align*}
y_t & \sim \text{Po}(\lambda_t) \\
\ln \lambda_t & = \mu_t + s_t + \gamma_t \\
\mu_t & = \mu_{t-1} + \omega_{1t} \\
s_t & = -(s_{t-1} + \ldots + s_{t-p+1}) + \omega_{2t} \\
\gamma_t & \sim \mathcal{N}(0, W_3),
\end{align*}
\]

where \(\omega_{it} \sim \mathcal{N}(0, W_i)\) independent for \(i = 1, 2\) and \(p = 12\) is the seasonal period.

We fit the seasonal Poisson SSM without and with overdispersion. First we take \(W_3 = 0\) which means \(\gamma_t = 0\) for all \(t\). Then, we take \(\gamma_t\) as described above such that the model can handle with overdispersion. We set the prior distributions as \(W_i^{-1} \sim \text{Gamma}(0.01, 0.01)\) for \(i = 1, 2, 3\), \(x_0 \sim \mathcal{N}(0, 10,000)\) and \(s_j \sim \mathcal{N}(0, 10,000)\) for \(j = -10, \ldots, 0\).
8.5.2.1 Computation Details

For MCMC, we use a single-move Gibbs sampler to draw values of the states $\lambda_t$ and $s_t$ in both cases. We run the MCMC algorithm for 700,000 iterations. We discard the first 100,000 iterations as a burn-in period. Next, in order to reduce the autocorrelation between successive values of the simulated chain, only every 100th value of the chain was stored.

For SMC, we ran the adaptive random walk Metropolis for 300,000 iterations and discarded the first 100,000, while the particle filter was run with 2,000 particles.

For INLA, the R commands required for the analysis are described in the Appendix.

8.5.2.2 Results

The results obtained with the different approximating schemes are summarized later in this chapter. They show reasonable agreement for both models between the different approximating schemes.

The main results for the system variances are summarized in Tables 8.3 and 8.4. Figure 8.4 shows the posterior smoothed mean of the mean rates $\lambda_t$ for model with overdispersion.

| TABLE 8.3 |
| Summary of the posterior distribution of the Poisson model without overdispersion applied to the Polio data set |
| Parameter | MCMC | SMC | INLA |
| $W_1$ | 0.0841 | 0.0889 | 0.0853 |
|  | (0.0201, 0.1953) | (0.0281, 0.2012) | (0.0273, 0.1939) |
| $W_2$ | 0.0209 | 0.0215 | 0.0206 |
|  | (0.0034, 0.0689) | (0.0039, 0.0679) | (0.0039, 0.0637) |

| TABLE 8.4 |
| Summary of the posterior distribution of the Poisson model with overdispersion applied to the Polio data set |
| Parameter | MCMC | SMC | INLA |
| $W_1$ | 0.0490 | 0.0472 | 0.0460 |
|  | (0.0089, 0.1485) | (0.0109, 0.1261) | (0.0107, 0.1187) |
| $W_2$ | 0.0162 | 0.0157 | 0.0146 |
|  | (0.0028, 0.0601) | (0.0032, 0.0502) | (0.0031, 0.0442) |
| $W_3$ | 0.2526 | 0.2648 | 0.2345 |
|  | (0.0185, 0.6822) | (0.0256, 0.6022) | (0.0319, 0.5339) |
8.6 Final Remarks

This chapter presents an overview of the possibilities associated with the analysis of discrete time series with SSM under the Bayesian perspective. A number of frequently used model components are described, including autoregressive dependence, seasonality, overdispersion, and transfer functions. The techniques most commonly used nowadays (MCMC, SMC, INLA) to analyze these models are also described and illustrated.

INLA is by far the fastest method for Bayesian data analysis, but suffers from a lack of measures to quantify and assess the errors committed. Also, it is not trivial to set the grid of values for computation of the integrals involved. This shortcoming is largely mitigated by the availability of the software R-INLA, but users become restricted to the options available there and are unable to easily introduce their own building blocks into the software.

SMC enables fairly fast generation of online posterior distributions of the state parameters and the hyperparameters. If one is interested in their smoothed distributions, as we showed in our illustrations, then their processing time becomes nonnegligible. This is still a lively area of research, and we may see in the near future ways to avoid the currently high computing cost of smoothed distributions. Some of the techniques associating SMC with MCMC seem to be promising.

Finally, MCMC is the standard approach for Bayesian data analysis these days. It is time consuming and requires nontrivial tuning when full conditionals are not available, but is well documented and has been used by many users under a large variety of settings.
Further, there are several reliable MCMC software that are equipped to handle a variety of sophisticated models. The results that we present in the two applications show some discrepancies between the methods. These are to be expected since there are errors involved in the approximations. However, we believe that these differences did not interfere with the overall analysis in any meaningful way. In summary, the message is that all these approaches can be safely used (especially in the linear realm) with the guidance provided in this chapter.

8A Appendix

This Appendix contains the commands required for the analysis of the datasets of Section 5 with the R package **INLA**.

8A.1 Deep Brain Stimulation

The first step is to load the R package **INLA** and set-up a data.frame that will be used within the inla() function.

```r
require(INLA)
# Create data.frame
data_frame = data.frame(y = y, z = z, time = 1:length(y))

head(data_frame, 3)
y z1 z2 time
1 0 1 1 1
2 1 1 1 2
3 0 3 1 3
```

The response variable \( y \) takes the value 1 to indicate a REM sleep cycle and zero otherwise. \( z_1 \) and \( z_2 \) stand for the \( z \) covariates described in Section 8.5.1 and \( \text{time} \) assumes the values \( t = 1, \ldots, n \), where \( n \) is the number of observations, and will be used to model the time-specific effect \( x_t \) on the observed process. The model can then be fitted with the command line

```r
inla1 <- inla(formula = y ~ 1 + z1 + z2 +
               f(time, model = "ar1", hyper = hyper.ar1),
               family = 'binomial',
               data = data_frame,
               control.fixed = control.fixed(prec.intercept=0.001,
                                              prec = 0.001),
               control.family = list(link = "probit"),
               control.predictor = list(compute = TRUE))
```
The formula argument of our model assumes that our linear predictor has an intercept, two covariates $z_1$ and $z_2$ and a time-specific effect indexed by $t$. The control.fixed argument was used to set the precision of the priors for the fixed-effects equal to 0.001, which leads to a variance equal to 1000. The model for the time-specific effect was set to follow an autoregressive model of order 1 and the priors for its hyperparameters $\delta$ and $\tau$ were set by $\text{hyper} = \text{hyper.ar1}$ inside the $f()$ function, where

$$\text{hyper.ar1} = \text{list}( \text{prec} = \text{list}(\text{prior} = \text{"loggamma"},\text{param} = \text{c}(0.01, 0.01)), \text{rho} = \text{list}(\text{prior} = \text{table})$$

and table was defined in such a way to encode $\delta \sim \mathcal{N}(-1,1)(0.95,100)$. For detailed information on how to set priors, please visit http://www.r-inla.org/models/priors.

The family argument indicates that we are dealing with binomial data. The control.family tells INLA that we want a probit link function. Lastly, besides the marginal posterior distribution of the latent field $x$ and the elements of hyperparameters $\theta$, control.predictor sets INLA to compute the marginal posterior distribution of the linear predictors $\eta(x)$ as well, which is not done by default.

### 8A.2 Poliomyelitis in the U.S.

Our data.frame contains the following columns: $y$, $time$, seasonal and iid, which represent the response variable $y_t$ and the indexes for the time-specific effect $\mu_t$, the seasonal effect $s_t$ and the i.i.d. random-effects $\gamma_t$, respectively.

```r
n = length(y)
data_frame = data.frame(y = y, time = 1:n,
seasonal = 1:n, iid = 1:n)
```

Notice that although $time$, seasonal, and iid assume identical values, the $\text{inla()}$ function require the definition of a different index for each random-effect defined through the $f()$ functions.

We first fit the model without accounting for overdispersion with command line

```r
n.seas = 12
formula1 = y ~ -1 +
  f(time, model = "rw1", constr=FALSE, hyper = hyper.rw1) +
  f(seasonal, model = "seasonal", season.length = n.seas,
    hyper = hyper.sea)
inla1 = inla(formula = formula1,
data = data_frame,
family="Poisson",
control.predictor = control.predictor(compute=TRUE))
```

In the model definition, $-1$ stands for the absence of an intercept in the model, the time-specific random-effects were chosen to follow a random-walk model of order 1 (rw1) without a sum-to-zero constrain $t$ (constr=FALSE) and for the seasonal effect we specified that we were interested in a monthly frequency by setting season.length = 12. The priors for the hyperparameter of the rw1 and the seasonal model were defined using a
similar syntax as the one displayed in Section 8.4.2. `family="Poisson"` indicates we are dealing with count data and that we will use a Poisson likelihood in our model.

We now proceed to a model that also includes random-effects to account for overdispersion. We have only redefined the arguments that change from one model to the other in order to save space, while the arguments that remain the same as in `inla1` have been represented by *. The command line becomes

```r
formula2 = y ~ -1 +
  f(time, model = "rw1", constr=FALSE, hyper = hyper.rw1) +
  f(seasonal, model = "seasonal", season.length = n.seas,
     hyper = hyper.sea) +
  f(iid, model = "iid", hyper = hyper.iid)
inla2 = inla(formula = formula2, *)
```

The only new feature of this model is the addition of the i.i.d. random-effects to account for overdispersion.

---

**Acknowledgments**

The authors thank an anonymous reviewer and the editors for their careful reading of this work, and for their comments and suggestions that helped to improve the presentation. Dani Gamerman was supported by a grant from CNPq-Brazil. Carlos A. Abanto-Valle was supported by grants from CNPq-Brazil and FAPERJ. Ralph S. Silva was supported by grants from CNPq-Brazil and FAPERJ.

---

**References**


