

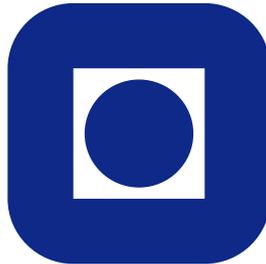
NORGES TEKNISK-NATURVITENSKAPELIGE  
UNIVERSITET

**Fitting dynamic models using integrated nested Laplace  
approximations – INLA**

by

Ramiro Ruiz-Cárdenas, Elias T. Krainski and Håvard Rue

PREPRINT  
STATISTICS NO. 12/2010



NORWEGIAN UNIVERSITY OF SCIENCE AND  
TECHNOLOGY  
TRONDHEIM, NORWAY

This preprint has URL <http://www.math.ntnu.no/preprint/statistics/2010/S12-2010.pdf>

Håvard Rue has homepage: <http://www.math.ntnu.no/~hrue>

E-mail: [hrue@math.ntnu.no](mailto:hrue@math.ntnu.no)

Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491  
Trondheim, Norway.

# Fitting dynamic models using integrated nested Laplace approximations - INLA

Ramiro Ruiz-Cárdenas<sup>†\*</sup>, Elias T. Krainski<sup>§</sup> and Håvard Rue<sup>‡</sup>

September 2, 2010

<sup>†</sup> Department of Statistics, Federal University of Minas Gerais – Belo Horizonte, Brazil

<sup>§</sup> Department of Statistics, Federal University of Paraná – Curitiba, Brazil

<sup>‡</sup> Norwegian University for Science and Technology – Trondheim, Norway

## 1 Introduction

State space models, also known as *dynamic models* in the Bayesian literature, are a broad class of parametric models with time varying parameters where both, the parameter variation and the available data information are described in a probabilistic way. They find application in the modeling and forecasting of time series data and regression (for a comprehensive treatment see for example West and Harrison, 1997; Durbin and Koopman, 2001). In this report we propose and illustrate through a series of examples, a computational framework to perform approximate inference in linear and generalized linear dynamic models based on the Integrated Nested Laplace Approximation (INLA) approach, which overcome some limitations of computational tools currently available in the dynamic modeling literature. INLA is a recent approach proposed by Rue and Martino (2007) and Rue et al. (2009) to perform fast Bayesian inference through the accurate approximation of the marginal posterior densities of hyperparameters and latent variables in latent Gaussian models. This class of statistical models embraces a wide range of models commonly used in applications, including generalized linear models, generalized additive models, smoothing spline models, semi-parametric regression, spatial and spatio-temporal models, log-Gaussian Cox processes, geostatistical and geosadditive models, besides state space models.

An efficient computational implementation of the procedures needed by the INLA approach was made in the open source library GMRFLib, a C-library for fast and exact simulation of Gaussian Markov random fields (Rue and Follstad, 2002). A user friendly interface for using INLA with the R programming language (R Development Core Team, 2010), hereinafter referred to as the INLA library, is also available from the web page <http://www.r-inla.org/>. Most of the latent models mentioned above have been successfully fitted using the INLA library and many examples are available from the INLA web page.

Currently just univariate dynamic models with a simple random walk evolution, such as first order and dynamic regression models, can be directly fitted by the INLA library. However, for more complex cases, such as growth models and spatio-temporal dynamic models, it is still possible to formulate specific latent models in a state-space form in order to perform approximate inference on them using INLA. The main aim of this report is to show how to

---

\*Corresponding author. Tel.: 55 31 34095905; fax: 55 31 34095924; e-mail: ramiro@est.ufmg.br

use the INLA library to perform inference in linear and generalized linear dynamic models. An inference framework to achieve this goal is proposed and illustrated with simulated and real life examples. A first approach uses existing model options in the INLA library to model random walk evolution and seasonal behavior of the different components of the dynamic models. A generic approach is also proposed to formulate and fit dynamic models in a more general setting, which is useful with more complex models, such as spatio-temporal dynamic models. This generic approach consists in merging the actual observations from the observational equation with “pseudo” observations coming from the evolution equations of the dynamic model in a unique structure and fit this augmented latent model in INLA considering different likelihoods for the observations and states. The combination of the two approaches is also possible. We show how this inference framework enables the fitting of several kinds of dynamic models, including realistically complex spatio-temporal models, in a short computational time and in a user friendly way.

The rest of the paper is organised as follows. In Section 2 we briefly introduce dynamic models and the main computational approaches in the literature to perform inference on this class of models. Section 3 describes the basics of the INLA computational approach. The proposed framework to fit dynamic models using INLA is illustrated in Section 4 through a series of simulated examples, firstly with the most common types of dynamic models and then with two cases of a complex spatio-temporal dynamic model. In Section 5 some well known worked examples from the literature are considered and their fitting using the INLA library is compared with current approaches. Concluding remarks and future work are stated in Section 6.

## 2 Dynamic models

According to Migon et al. (2005), dynamic models can be seen as a generalization of regression models, allowing changes in parameter values throughout time by the introduction of an equation governing the temporal evolution of regression coefficients. In the Gaussian case they consist on the couple of equations

$$y_t = F_t' X_t + \nu_t, \quad \nu_t \sim N(0, V_t) \quad (1)$$

$$X_t = G_t X_{t-1} + \omega_t, \quad \omega_t \sim N(0, W_t), \quad (2)$$

where  $y_t$  is a time sequence of scalar observations,  $X_t$  is a sequence of state (latent) parameters describing locally the system. It is assumed that  $y_t$  is conditionally independent given  $X_t$ .  $F_t$  is a vector of explanatory variables and  $G_t$  is a matrix describing the states evolution. The disturbances  $\nu_t$  and  $\omega_t$  are assumed to be both serially independent and also independent of each other. Therefore, the model is completely specified by the quadruple  $\{F_t; G_t; V_t; W_t\}$ . When these quantities are known, inference on the states  $X_t$  can be performed analytically through an iterative procedure using the Kalman filter algorithm (for details see for example West and Harrison, 1997).

On the other hand, when the hyperparameters  $V_t$  and  $W_t$  are unknown, inference in dynamic linear models is not available analytically. In order to circumvent this problem, several proposals to perform approximate inference in DLMS have appeared in the literature. Early work includes approaches based on the extended Kalman filter (Jazwinski, 1970) or on Gaussian quadratures (Pole and West, 1990). In recent years, considerable attention has been concentrated in simulation-based approaches, such as sequential Monte Carlo also known as particle filters (e.g., Gordon et al., 1993; Godsill et al., 2004) and especially Markov chain Monte Carlo

(MCMC) methods. The later is currently the most common approach to inference in dynamic models (e.g., Gamerman, 1998; Reis et al., 2006) due to its generality and capability to obtain samples from the the posterior distribution of all unknown model parameters in an efficient way. However, MCMC implementation is more involved and it suffers from a series of well known problems that have hindered its wider utilization in applied settings. For example, convergence can be quite difficult to diagnose and the computational cost may become prohibitively high for complex models, as is the case of spatio-temporal dynamic models. The Monte Carlo errors are also intrinsically large and strong correlation among parameters is common, leading the algorithms slow.

A sort of computational tools to fit dynamic models using some of the inference methods mentioned above have also appeared in the literature to aid end users to benefit from methodological developments. The first of them was the `Bats` software (West et al., 1988; Pole et al., 1994), a package for time series analysis and forecasting using Bayesian dynamic modeling developed in the late 80's by the "Bayesian Forecasting Group" of Warwick University. It deals with univariate time series and dynamic regression models. It performs sequential estimation and uses a discount factor approach to model the unknown variances.

The `SsfPack` library (Koopman et al., 1999), a module for the programming language `Ox`, provides functions for likelihood evaluation and signal extraction of linear Gaussian state space models, with support for estimating some non-Gaussian and nonlinear models using importance sampling and Markov chain Monte Carlo (MCMC) methods.

More recently some R packages and functions to fit dynamic linear and generalized linear models have been developed. The function `StructTS` written by Bryan Ripley (see Ripley, 2002) fits linear Gaussian state-space models (also called structural models) for univariate time series by maximum likelihood, by decomposing the series in trend and/or seasonal components. The `dlm` package (Petris, 2010), performs maximum likelihood, Kalman filtering and smoothing, and Bayesian analysis of Gaussian linear dynamic models. It applies Kalman filter to compute filtered values of the state vectors, together with their variance/covariance matrices. The calculations are based on the singular value decomposition (SVD) of the relevant matrices. Variance matrices are returned in terms of their SVD. It also implements a Gibbs sampler for a univariate DLM having one or more unknown variances in its specification.

The `sspir` package (Dethlefsen and Lundbye-Christensen, 2006) includes functions for Kalman filtering and smoothing of dynamic linear and generalized linear models. Estimation of variance matrices can be performed using the EM algorithm in the Gaussian case, but it requires that the variance matrices to be estimated are constant. Non-Gaussian state space models are approximated to a Gaussian state-space model through an iterative procedure based on an iterated extended Kalman filter. This procedure is for the calculation of the conditional mean and variance of the latent process (not for the hyperparameters). The `KFAS` package (Helske, 2010), also provides functions for Kalman filtering and smoothing of univariate exponential family state space models. It allows diffuse initialization when distributions of some or all elements of initial state vector are unknown.

These computational tools have significantly contributed to a wider use of dynamic models in applied settings. However, support remains incomplete in some particular aspects of dynamic modeling. For example, the above approaches in general allow missing values just in the observation vector, but not in the covariates. As we leave the Gaussian univariate case, estimation of hyperparameters and its uncertainty is not as straightforward to obtain. Estimation of more complex dynamic models as is the case of the spatio-temporal ones is not possible with these tools. In the next sections we show how the INLA approach can be used to fill these gaps in a simple and flexible way.

### 3 The Integrated Nested Laplace Approximation (INLA) approach

INLA is a computational approach recently introduced by Rue et al. (2009) to perform Bayesian inference in the broad class of latent Gaussian models, that is, models of an outcome variable  $y_i$  that assume independence conditional on some underlying (unknown) latent field  $\boldsymbol{\xi}$  and a vector of hyperparameters  $\boldsymbol{\theta}$ . It was proposed as an alternative to the usually time consuming MCMC methods. Unlike MCMC where posterior inference is sample-based, the INLA computational approach approximates the posteriors of interest with a closed form expression. Therefore, problems of convergence and mixing, inherent to MCMC runs, are not an issue. The main aim of the INLA approach is to approximate the marginal posteriors for the latent variables as well as for the hyperparameters of the Gaussian latent model, given by

$$\pi(\xi_i | \mathbf{Y}) = \int \pi(\xi_i | \boldsymbol{\theta}, \mathbf{Y}) \pi(\boldsymbol{\theta} | \mathbf{Y}) d\boldsymbol{\theta} \quad (3)$$

$$\pi(\theta_j | \mathbf{Y}) = \int \pi(\boldsymbol{\theta} | \mathbf{Y}) d\theta_{-j}. \quad (4)$$

This approximation is based on an efficient combination of (analytical Gaussian) Laplace approximations to the full conditionals  $\pi(\boldsymbol{\theta} | \mathbf{Y})$  and  $\pi(\xi_i | \boldsymbol{\theta}, \mathbf{Y})$ ,  $i = 1, \dots, n$ , and numerical integration routines to integrate out the hyperparameters  $\boldsymbol{\theta}$ .

The INLA approach as proposed in Rue et al. (2009) includes three main approximation steps to obtain the marginal posteriors in (3) and (4). The first step consists in approximate the full posterior  $\pi(\boldsymbol{\theta} | \mathbf{Y})$ . To achieve this, firstly an approximation to the full conditional distribution of  $\boldsymbol{\xi}$ ,  $\pi(\boldsymbol{\xi} | \mathbf{Y}, \boldsymbol{\theta})$  is obtained using a multivariate Gaussian density  $\tilde{\pi}_G(\boldsymbol{\xi} | \mathbf{Y}, \boldsymbol{\theta})$  (for details see Rue and Held, 2005) and evaluated at its mode. Then the posterior density of  $\boldsymbol{\theta}$  is approximated by using the Laplace approximation

$$\tilde{\pi}(\boldsymbol{\theta} | \mathbf{Y}) \propto \frac{\pi(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})}{\tilde{\pi}_G(\boldsymbol{\xi} | \boldsymbol{\theta}, \mathbf{Y})} \Big|_{\boldsymbol{\xi}=\boldsymbol{\xi}^*(\boldsymbol{\theta})},$$

where  $\boldsymbol{\xi}^*(\boldsymbol{\theta})$  is the mode of the full conditional of  $\boldsymbol{\xi}$  for a given  $\boldsymbol{\theta}$ . Since no exact closed form is available for  $\boldsymbol{\xi}^*(\boldsymbol{\theta})$ , an optimization scheme is necessary. Rue et al. (2009) computes this mode using the Newton-Raphson algorithm. The posterior  $\tilde{\pi}(\boldsymbol{\theta} | \mathbf{Y})$  will be used later to integrate out the uncertainty with respect to  $\boldsymbol{\theta}$  when approximating the posterior marginal of  $\xi_i$ .

The second step computes the Laplace approximation of the full conditionals  $\pi(\xi_i | \mathbf{Y}, \boldsymbol{\theta})$  for selected values of  $\boldsymbol{\theta}$ . These values will be used as evaluation points in the numerical integration applied to obtain the posterior marginals of  $\xi_i$  in (3). The density  $\pi(\xi_i | \boldsymbol{\theta}, \mathbf{Y})$  is approximated using the Laplace approximation defined by:

$$\tilde{\pi}_{LA}(\xi_i | \boldsymbol{\theta}, \mathbf{Y}) \propto \frac{\pi(\boldsymbol{\xi}, \boldsymbol{\theta}, \mathbf{Y})}{\tilde{\pi}_G(\boldsymbol{\xi}_{-i} | \xi_i, \boldsymbol{\theta}, \mathbf{Y})} \Big|_{\boldsymbol{\xi}_{-i}=\boldsymbol{\xi}_{-i}^*(\xi_i, \boldsymbol{\theta})}, \quad (5)$$

where  $\boldsymbol{\xi}_{-i}$  denotes the vector  $\boldsymbol{\xi}$  with the  $i$ th component omitted,  $\tilde{\pi}_G(\boldsymbol{\xi}_{-i} | \xi_i, \boldsymbol{\theta}, \mathbf{Y})$  is the Gaussian approximation of  $\pi(\boldsymbol{\xi}_{-i} | \xi_i, \boldsymbol{\theta}, \mathbf{Y})$ , treating  $\xi_i$  as fixed (observed) and  $\boldsymbol{\xi}_{-i}^*(\xi_i, \boldsymbol{\theta})$  is the mode of  $\pi(\boldsymbol{\xi}_{-i} | \xi_i, \boldsymbol{\theta}, \mathbf{Y})$ .

The approximation of  $\pi(\xi_i | \boldsymbol{\theta}, \mathbf{Y})$  using (5) can be quite expensive, since  $\tilde{\pi}_G(\boldsymbol{\xi}_{-i} | \xi_i, \boldsymbol{\theta}, \mathbf{Y})$  must be recomputed for each value of  $\xi_i$  and  $\boldsymbol{\theta}$ . Two alternatives are proposed in Rue et al. (2009) to obtain these full conditionals in a cheapest way. The first one is just the Gaussian approximation  $\tilde{\pi}_G(\xi_i | \boldsymbol{\theta}, \mathbf{Y})$ , which provides reasonable results in short computational time.

However, according to Rue and Martino (2007), its accuracy can be affected by errors in the location and/or errors due to the lack of skewness. The second alternative uses a simplified version of the Laplace approximation,  $\tilde{\pi}_{SLA}(\xi_i | \boldsymbol{\theta}, \mathbf{Y})$ , defined as the series expansion of  $\tilde{\pi}_{LA}(\xi_i | \boldsymbol{\theta}, \mathbf{Y})$  around  $\xi_i = \mu_i(\boldsymbol{\theta})$  (for details see Rue et al., 2009).

Finally, in the third step the full posteriors obtained in the previous two approximation steps are combined and the marginal posterior densities of  $\xi_i$  and  $\theta_j$  are obtained by integrating out the irrelevant terms. The approximation for the marginal of the latent variables can be obtained by the expression

$$\pi(\xi_i | \mathbf{Y}) = \int \pi(\xi_i | \mathbf{Y}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \mathbf{Y}) d\boldsymbol{\theta} \approx \sum_k \tilde{\pi}(\xi_i | \theta_k, \mathbf{Y}) \tilde{\pi}(\theta_k | \mathbf{Y}) \Delta_k, \quad (6)$$

which is evaluated using numerical integration on a set  $\theta_k$  of grid points, with area weights  $\Delta_k$  for  $k = 1, 2, \dots, K$ . According to Rue et al. (2009), since the points  $\theta_k$  are selected in a regular grid, it is feasible to take all the area weights  $\Delta_k$  to be equal. In a similar way, the posterior approximation  $\tilde{\pi}(\boldsymbol{\theta} | \mathbf{Y})$  is explored by numerical integration routines for evaluation of the marginal

$$\pi(\theta_j | \mathbf{Y}) = \int \pi(\boldsymbol{\theta} | \mathbf{Y}) d\theta_{-i} \approx \int \tilde{\pi}(\theta_k | \mathbf{Y}) d\theta_{-i}.$$

Since the dimension of  $\boldsymbol{\theta}$  is assumed small (i.e.,  $\leq 7$ ), these numerical routines are efficient in returning a discretized representation of the marginal posteriors.

A good choice of the set  $\theta_k$  of evaluation points is crucial to the accuracy of the above numerical integration steps. In order to do that, Rue et al. (2009) suggest to compute the negative Hessian matrix  $S$  at the mode,  $\theta^*$ , of  $\tilde{\pi}(\boldsymbol{\theta} | \mathbf{Y})$  and to consider its spectral value decomposition,  $S^{-1} = Q\Lambda Q^T$ . Then a standardized variable  $z$  is defined as

$$z = Q^T \Lambda^{-1/2} (\boldsymbol{\theta} - \theta^*) \quad \text{or} \quad \boldsymbol{\theta}(z) = \theta^* + Q\Lambda^{1/2} z$$

and a collection,  $Z$ , of  $z$  values is found, such that the corresponding  $\boldsymbol{\theta}(z)$  points are located around the mode  $\boldsymbol{\theta} = \theta^*$ . Starting from  $z = 0$  ( $\boldsymbol{\theta} = \theta^*$ ), each component entry of  $z$  is searched in the positive and negative directions in step sizes of  $\eta_z$ . All  $z$ -points satisfying

$$\log \tilde{\pi}(\boldsymbol{\theta}(0) | \mathbf{Y}) - \log \tilde{\pi}(\boldsymbol{\theta}(z) | \mathbf{Y}) < \eta_\pi$$

are taken to be in  $Z$ . The set  $\theta_k$  of evaluation points is finally based on the values in  $Z$ . An appropriate tuning of the  $\eta_z$  and  $\eta_\pi$  values should be performed in order to produce accurate approximations.

The following two sections show, through a series of examples, how the INLA approach can be extended to deal with inference in dynamic models in an easy way using the INLA library.

## 4 Simulated examples

In this section we generate some simulated data sets in order to illustrate the formulation of the more common types of dynamic models using the INLA library. The steps to perform inference on these models with INLA using the R programming language (R Development Core Team, 2010) are described in detail using a simple univariate linear dynamic model. The methodology is then applied on a second order polynomial dynamic model, a seasonal dynamic model and a Poisson dynamic regression model. Finally two cases of a complex spatio-temporal dynamic model for Gaussian areal data are considered, showing the capability of INLA to fit realistically complex dynamic models.

### Example 1: A toy example

We begin with a very simple simulated example of a first order univariate dynamic linear model in order to gain insight into the specification of dynamic models for use within INLA. For details of how to simulate observations from this and the other models in this section see the R script accompanying this report. The model has the following observational and system equations:

$$y_t = X_t + \nu_t, \quad \nu_t \sim N(0, V), \quad t = 1, \dots, n \quad (7)$$

$$X_t = X_{t-1} + \omega_t, \quad \omega_t \sim N(0, W), \quad t = 2, \dots, n \quad (8)$$

Since the evolution of states in this simple model follows a first order random walk process, it could be fitted with the INLA library just using model option “rw1” according to the following code:

```
i <- 1:n          # indices for x_t
ww <- rep(1,n)    # weights for x_t
formula <- y ~ ww + f(i, ww, model="rw1") -1
r <- inla(formula, data = data.frame(i,ww,y))
```

However, we will use this simple example to illustrate a generic approach to fit dynamic models in INLA based on an augmented model structure, which will be useful later on with more complex models. The key to fit this model with the INLA library under this approach consists in equating to zero the system equation, that is, we re-write (8) as

$$0 = X_t - X_{t-1} + \omega_t, \quad t = 2, \dots, n$$

and then we build an augmented model with dimension  $n + (n - 1)$  merging these “faked zero observations” from the system equation with the actual observations from the observational equation in a unique structure, as shown in Figure 4, where the first  $n$  elements correspond to the  $n$  actual observations,  $\mathbf{y}_t = \{y_1, \dots, y_n\}$ , while the remaining  $n - 1$  elements (corresponding to the number of state parameters in Eq. 8) are forced to be zero.

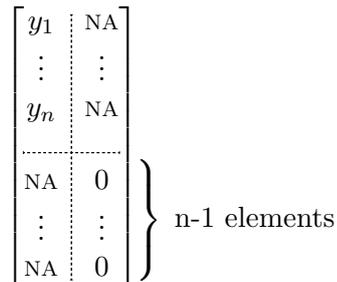


Figure 4. Schematic representation of the data structure for the augmented model.

Inference in this augmented model using the INLA approach is performed considering two different likelihoods. The first  $n$  data points are assumed to follow a Gaussian distribution with unknown precision  $V^{-1}$ , whereas the last  $n - 1$  data points, which are forced to be zero, are considered as observed with a high and fixed precision. Let  $\mathbf{y}$  be a  $n \times 1$  vector of observations simulated from the dynamic linear model described above. The following R code fits this model using the INLA library:

```
# building the augmented model
m <- n-1
Y <- matrix(NA, n+m, 2)
```

```

Y[1:n,      1] <- y                # actual observations
Y[1:m + n, 2] <- 0                # faked observations

# indices for the INLA library
i <- c(1:n, 2:n)                  # indices for x_t
j <- c(rep(NA,n), 2:n -1)        # indices for x_{t-1}
w1 <- c(rep(NA,n), rep(-1,m))    # weights for x_{t-1}
l <- c(rep(NA,n), 1:m)           # indices for w_t

# formulating the model
formula <- Y ~ f(i, model="iid", initial=-10, fixed=T) +
           f(j, w1, copy="i") +
           f(l, model ="iid") -1

# call to fit the model
require(INLA)
r <- inla(formula, data = data.frame(i,j,w1,l),
           family = rep("gaussian", 2),
           control.data = list(list(), list(initial=10, fixed=T)))

```

In order to be properly considered by the INLA library, each term on the right side of the observational and system equations must be indexed, according to its corresponding time index, in the dataframe to be passed to INLA. For example, the state vector  $X_t$  appears in Eq. (7) at times  $t = 1, 2, \dots, n$  as well as in Eq. (8) at times  $t = 2, 3, \dots, n$ . Therefore, the index vector for the  $X_t$  term in the augmented dataframe is specified as  $[1, 2, \dots, n, 2, 3, \dots, n]'$ . Note that each column in the dataframe passed to INLA must be a vector of dimension equal to the number of rows of the augmented structure, which is  $n + (n - 1)$  in this example. When a term is present in just one part of the augmented structure, the remaining elements of the index vector are filled in with NA's. This is the case of term  $X_{t-1}$ , which appears just in Eq. (8). Its index vector then will be given by  $[NA, \dots, NA, 2, 3, \dots, n]'$ , where the first  $n$  elements are NA's.

Note that for the formulation of  $X_t$  we just use `model='iid'` with a fixed low precision. Figure 1 shows the series of simulated and estimated values for the vectors of observations and states respectively. Fit was quite good in both cases.

The model was fitted three times with different log-gamma priors for  $\log(V^{-1})$  and  $\log(W^{-1})$  specified as follows: an informative prior, with mean equal to the real value and coefficient of variation equal to 0.5, a vague prior also centered on the true simulated value but with coefficient of variation equal to 10 and the default INLA log-gamma prior. Precisions of the perturbation terms in model 1 were also well estimated as can be seen in Figure 2. Credibility intervals in all cases included the true simulated values:

	mean	sd	0.025quant	0.975quant
Precision for Gaussian observations	1.231873	0.2738968	0.7782617	1.848884
Precision for w_t	1.844574	0.5945417	0.9352658	3.247046

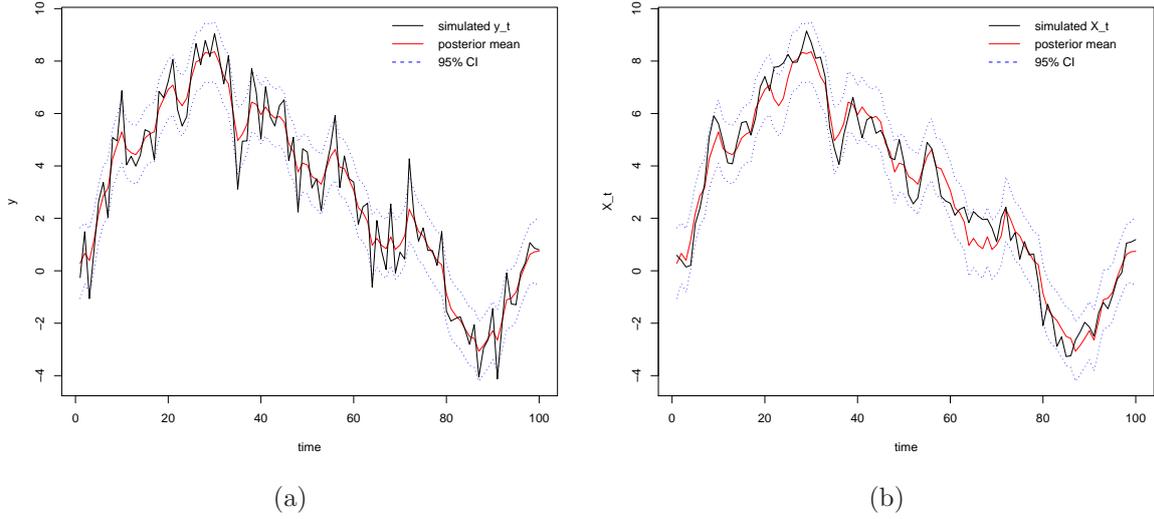


Figure 1: Simulated and predicted values (posterior mean and 95% credibility interval) for the observations (a) and states (b) in the toy example.

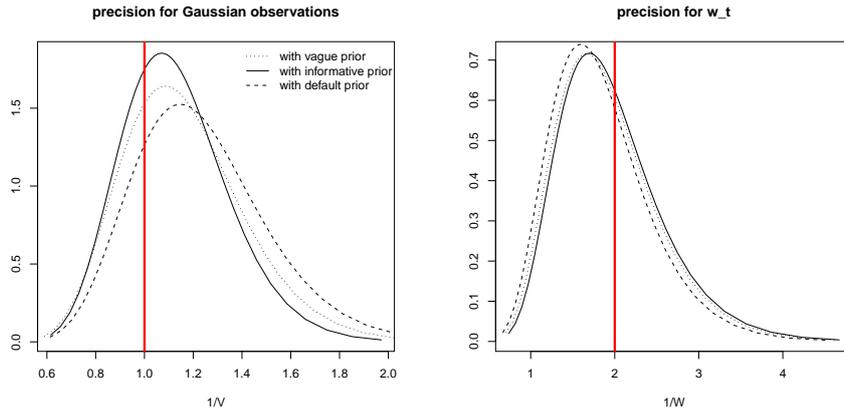


Figure 2: Posterior densities for the hyperparameters in the toy example. Red lines indicate true simulated values.

## Example 2: A second order polynomial DLM

The next simulated example corresponds to a second order polynomial dynamic model. In this case the state vector comprises two elements,  $X_t = (X_{1t}, X_{2t})$ , the first representing the current level and the second representing the current rate of change in the level. The response again is assumed to be normal. The observational and system equations for this model are given by

$$\begin{aligned}
 y_t &= X_{1t} + \nu_t, & \nu_t &\sim N(0, V), & t &= 1, \dots, n & (9) \\
 X_{1t} &= X_{1,t-1} + X_{2,t-1} + \omega_{1t}, & \omega_{1t} &\sim N(0, W_1), & t &= 2, \dots, n & (10) \\
 X_{2t} &= X_{2,t-1} + \omega_{2t}, & \omega_{2t} &\sim N(0, W_2), & t &= 2, \dots, n & (11)
 \end{aligned}$$

The augmented model in this case after merge the faked observations from equations (10) and (11) with the actual observations from equation (9), has dimension  $n + 2(n - 1)$  and three different likelihoods (see Figure 4), being the first  $n$  elements Gaussian distributed with unknown precision  $V$  and the remaining  $2(n - 1)$  elements forced to be zero and considered as observed with a high and fixed precision.

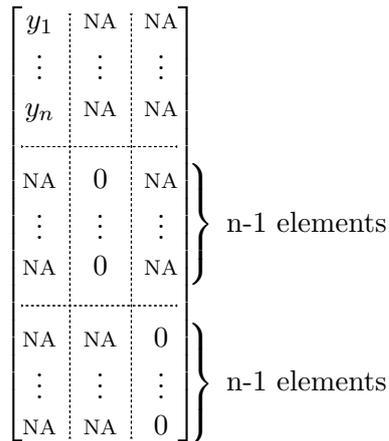
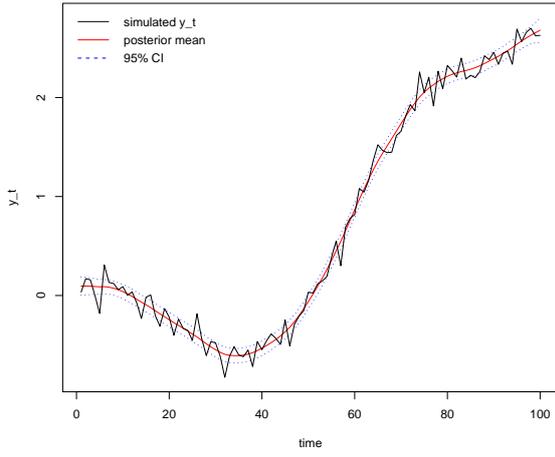


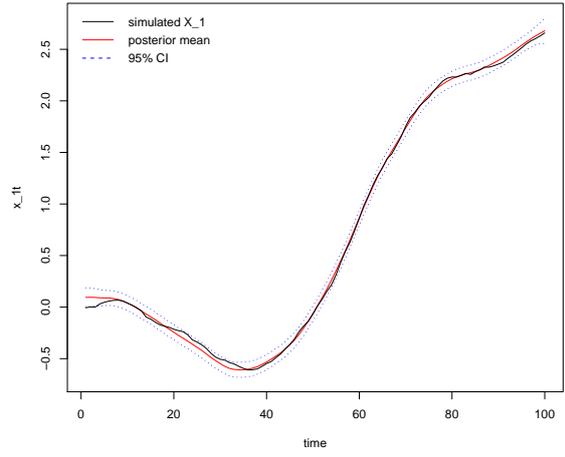
Figure 4. Schematic representation of the data structure for the augmented model in the second order DLM.

Simulated and predicted values for observations and states are presented in Figure 3. The posterior densities for the three precision parameters are also shown in Figure 4. Simulated values for  $V^{-1}$   $W_1^{-1}$  and  $W_2^{-1}$  in this example were 100, 10000 and 10000, respectively.

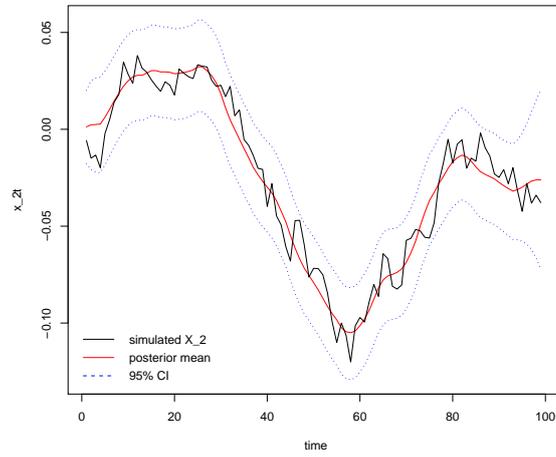
	mean	sd	0.025quant	0.5quant	0.975quant
Precision for observations	92.1002	13.8551	67.9370	91.0290	122.3182
Precision for w_1	20455.6903	19013.7524	2080.9327	15023.2395	70370.9427
Precision for w_2	2345.8005	15874.0193	5184.7682	17947.9580	65789.3177



(a)



(b)



(c)

Figure 3: Simulated and predicted values (posterior mean and 95% credibility interval) for the observations (a) and for  $X_{1t}$  (b) and  $X_{2t}$  (c) state vectors in the second order polynomial DLM.

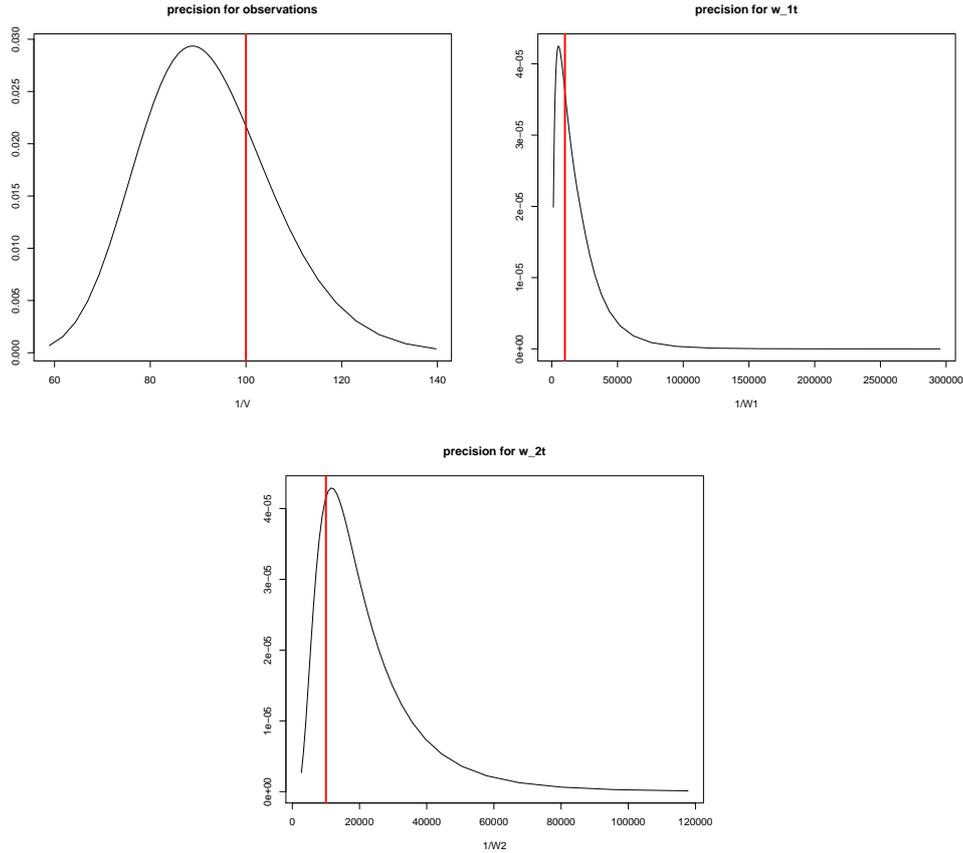


Figure 4: Posterior densities for the hyperparameters in the second order polynomial DLM. Red lines indicate true simulated values.

### Example 3: A seasonal DLM with harmonics

In this example we simulate a monthly time series with an annual seasonal pattern by using firstly a cosine form and then a sum of sine and cosine terms with the same frequency. The response again is assumed to be normal. The DLM formulation for the first case is defined as follows:

$$\begin{aligned}
 y_t &= a_t \cos\left(\frac{\pi(t-1)}{6}\right) + \nu_t, & \nu_t &\sim N(0, V), & t &= 1, \dots, n \\
 a_t &= a_{t-1} + \omega_{1t}, & \omega_{1t} &\sim N(0, W_1), & t &= 2, \dots, n
 \end{aligned} \tag{12}$$

This model can easily be fitted in INLA just using model option “RW1” as follows:

```

t <- 1:n
cosw <- cos(pi*(t-1)/6)
i <- 1:n
formula <- y ~ cosw + f(i, cosw, model="rw1") -1
r <- inla(formula, data = data.frame(i,y),
          control.predictor=list(compute=TRUE))

```

The results are shown in Figures 5 and 6:

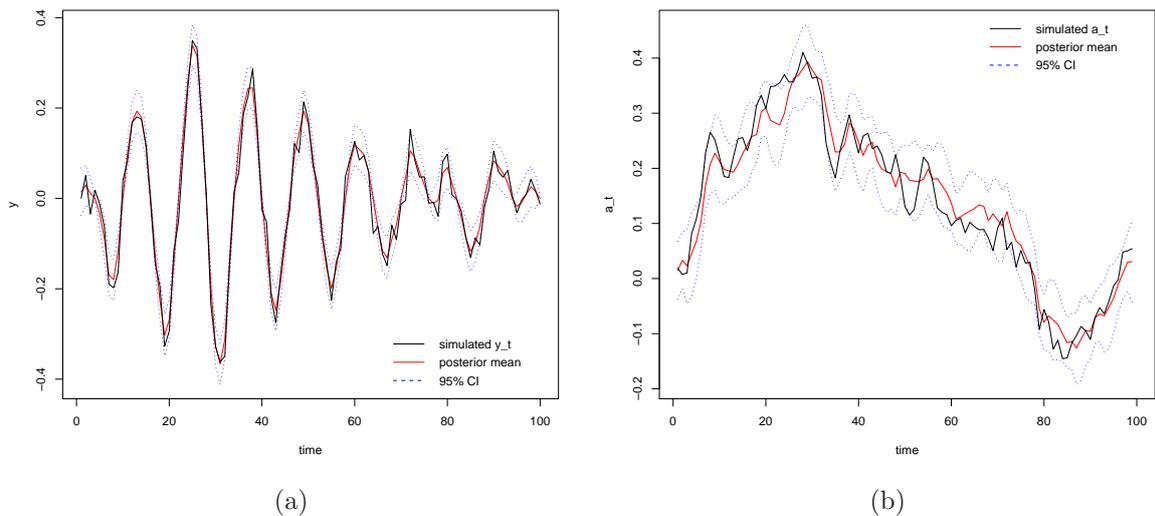


Figure 5: Simulated and predicted values (posterior mean and 95% credibility interval) for the observations (a) and for  $a_t$  state vector (b) in the first seasonal dynamic model.

The posterior densities for the precision parameters were:

	mean	sd	0.025quant	0.5quant	0.975quant
Precision for observations	1022.7902	229.5974	642.4997	998.8874	1540.3168
Precision for $w_{1t}$	428.5013	121.7585	238.3670	411.8994	713.8945

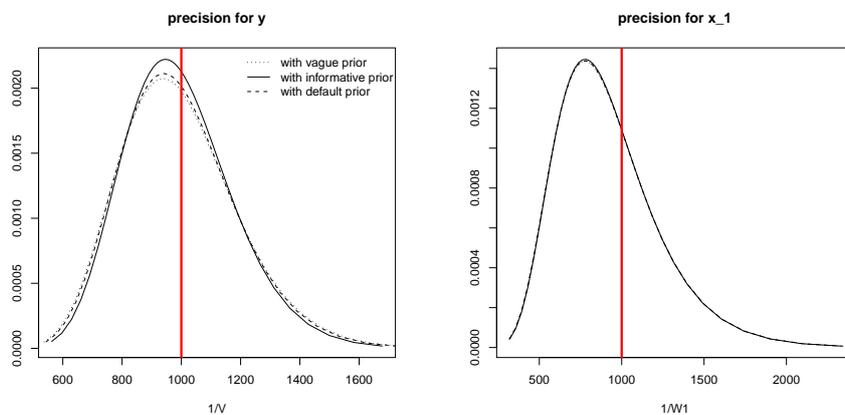


Figure 6: Posterior densities for the hyperparameters in the first seasonal dynamic model. Red lines indicate true simulated values.

Now we will simulate a monthly time series with an annual seasonal pattern by using a sum of sine and cosine terms with the same frequency. This model has a two parameter state  $X_t = (a_t, b_t)$ , where  $a_t$  models an arbitrary amplitude (seasonal peak to trough variation) and  $b_t$  allows the cycle maximum and minimum to be phase shifted. The DLM formulation for this model is defined as follows:

$$X_t = \begin{pmatrix} a_t \\ b_t \end{pmatrix}; \quad F_t = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad G_t = \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix}; \quad \text{where } \phi = \frac{\pi}{6}.$$

Therefore, the observational and system equations are given by

$$\begin{aligned}
 y_t &= a_t + \nu_t, & \nu_t &\sim N(0, V), & t &= 1, \dots, n \\
 a_t &= \cos(\phi)a_{t-1} + \sin(\phi)b_{t-1} + \omega_{1t}, & \omega_{1t} &\sim N(0, W_1), & t &= 2, \dots, n \\
 b_t &= -\sin(\phi)a_{t-1} + \cos(\phi)b_{t-1} + \omega_{2t}, & \omega_{2t} &\sim N(0, W_2), & t &= 2, \dots, n
 \end{aligned}
 \tag{13}$$

The augmented structure approach is necessary in this example. Equating to zero the system equations (13) and (14) and merging them with the observational equation yields an augmented model of dimension  $n + 2(n - 1)$ , which is fitted considering three different likelihoods. Results are shown in Figures 7 and 8:

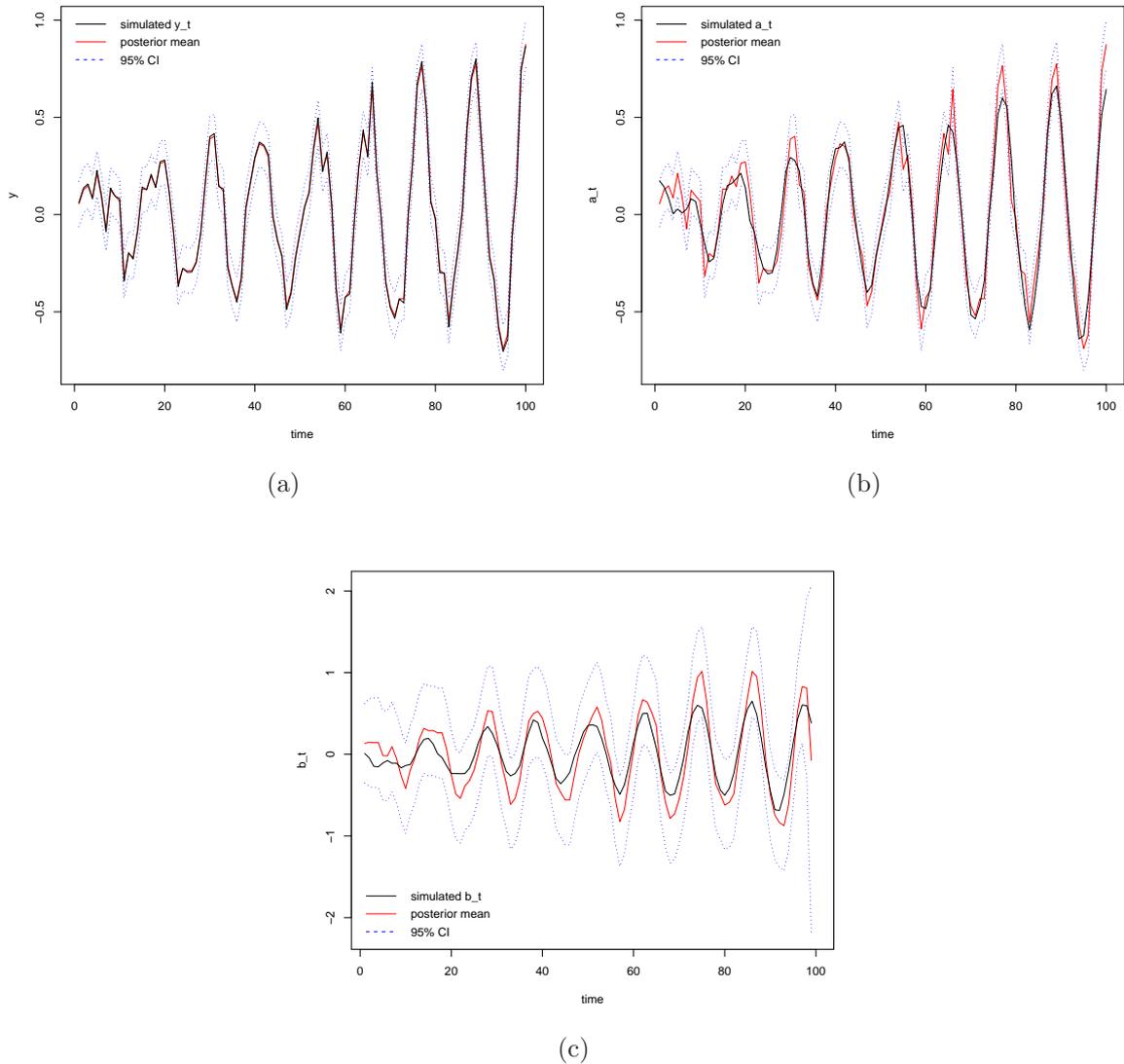


Figure 7: Simulated and predicted values (posterior mean and 95% credibility interval) for the observations (a) and for  $a_t$  (b) and  $b_t$  (c) state vectors in the second seasonal dynamic model.

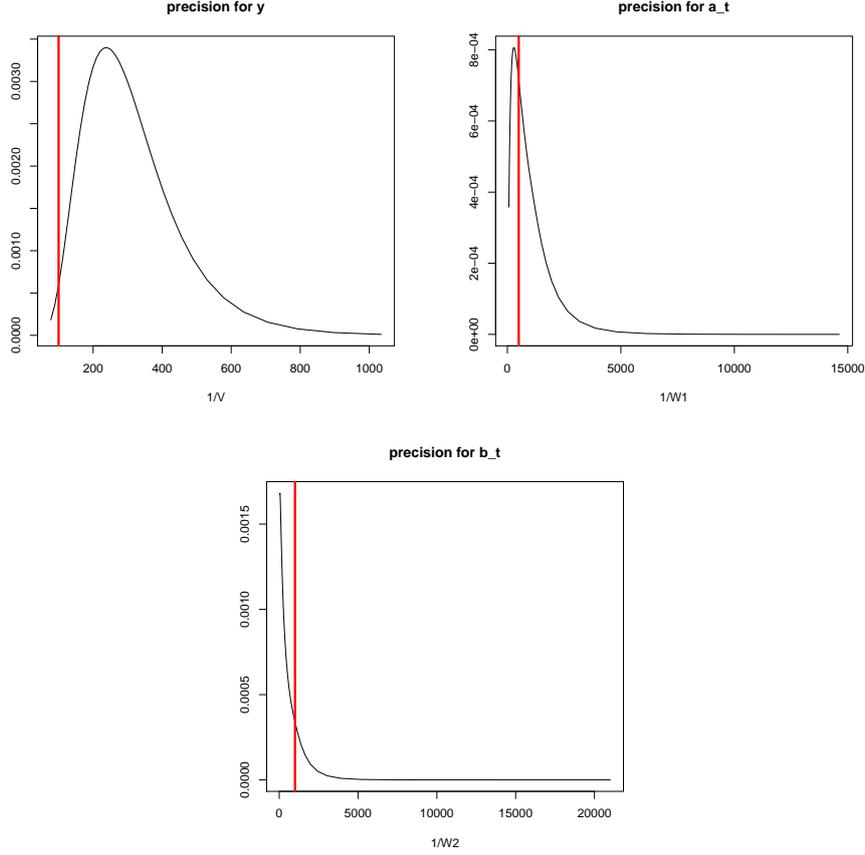


Figure 8: Posterior densities for the hyperparameters in the second seasonal dynamic model. Red lines indicate true simulated values.

#### Example 4: A Poisson dynamic multiple regression

Here we simulated data from a multiple Poisson regression model with two regressors,  $Z_{1t}$  and  $Z_{2t}$ . Therefore, the linear predictor is given by  $\lambda_t = \mathbf{F}_t \mathbf{x}_t$ , where  $\mathbf{F}_t = (1, Z_{1t}, Z_{2t})$  and the regression coefficients  $\mathbf{x}_t = (\beta_{0t}, \beta_{1t}, \beta_{2t})$  follow a simple random walk evolution. The model has the following observational and system equations:

$$(y_t | \mu_t) \sim \text{Poisson}(\mu_t)$$

$$\log(\mu_t) = \lambda_t = \beta_{0t} + \beta_{1t}Z_1 + \beta_{2t}Z_2 \quad t = 1, \dots, n$$

$$\beta_{0t} = \beta_{0,t-1} + \omega_{0t}, \quad \omega_{0t} \sim N(0, W_0), \quad t = 2, \dots, n \quad (15)$$

$$\beta_{1t} = \beta_{1,t-1} + \omega_{1t}, \quad \omega_{1t} \sim N(0, W_1), \quad t = 2, \dots, n \quad (16)$$

$$\beta_{2t} = \beta_{2,t-1} + \omega_{2t}, \quad \omega_{2t} \sim N(0, W_2), \quad t = 2, \dots, n \quad (17)$$

Since all regression coefficients in this model follow a simple random walk evolution, the augmented structure is not necessary. Instead, we simply use model “RW1” for all regression coefficients. This is achieved in R with the following code:

```
id <- id1 <- id2 <- 1:n

formula <- y ~ f(id, model="rw1",param=c(1,0.001),initial=5) +
  x1 + f(id1, x1, model="rw1",param=c(1,0.01),initial=5) +
  x2 + f(id2, x2, model="rw1",param=c(1,0.001),initial=5)
```

```

require(INLA)
r = inla(formula, family="poisson", data = data.frame(id,id1,id2,y),
        control.predictor=list(compute=TRUE))

```

Results are shown in Figures 9 and 10 :

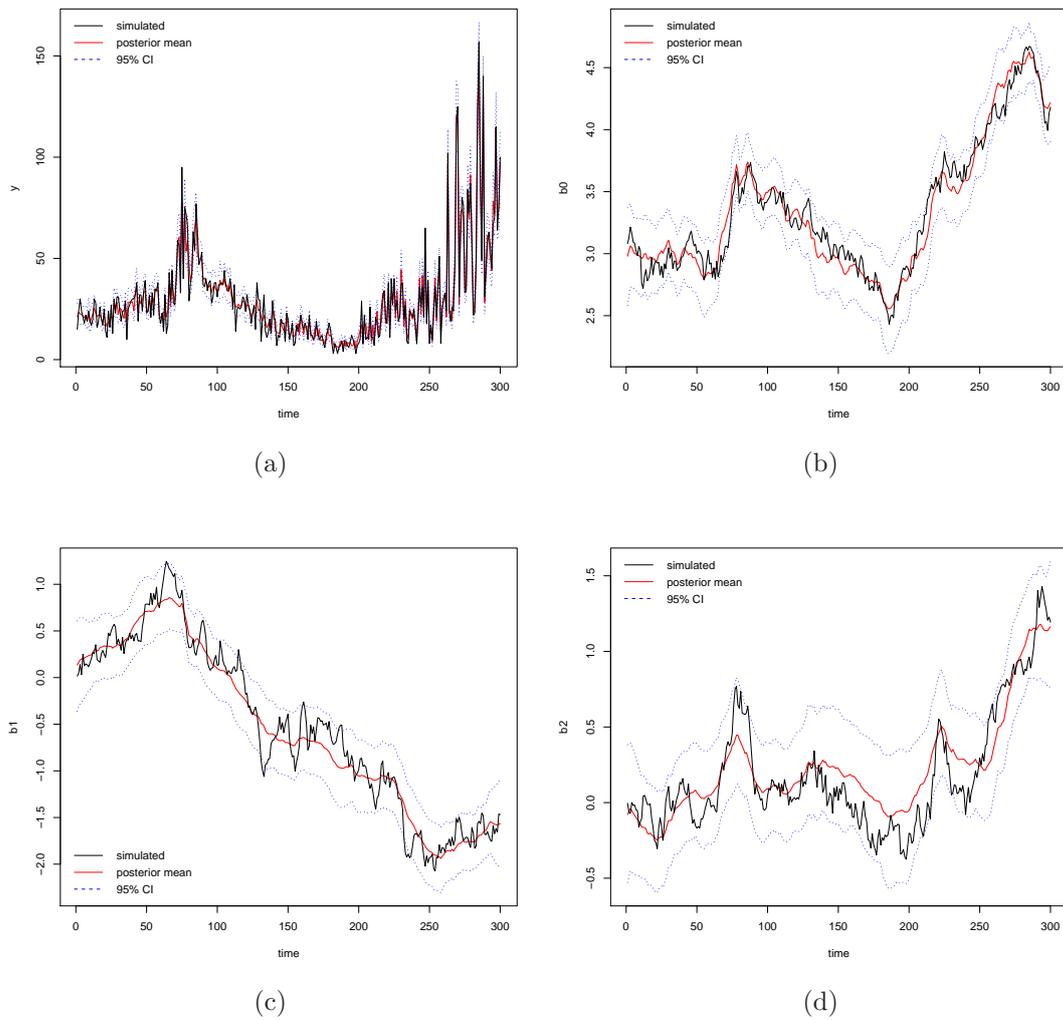


Figure 9: Simulated and predicted values (posterior mean and 95% credibility interval) for the observations (a) and regression coefficients,  $\beta_0$ ,  $\beta_1$  and  $\beta_2$  (b-d) in the multiple Poisson dynamic regression example.

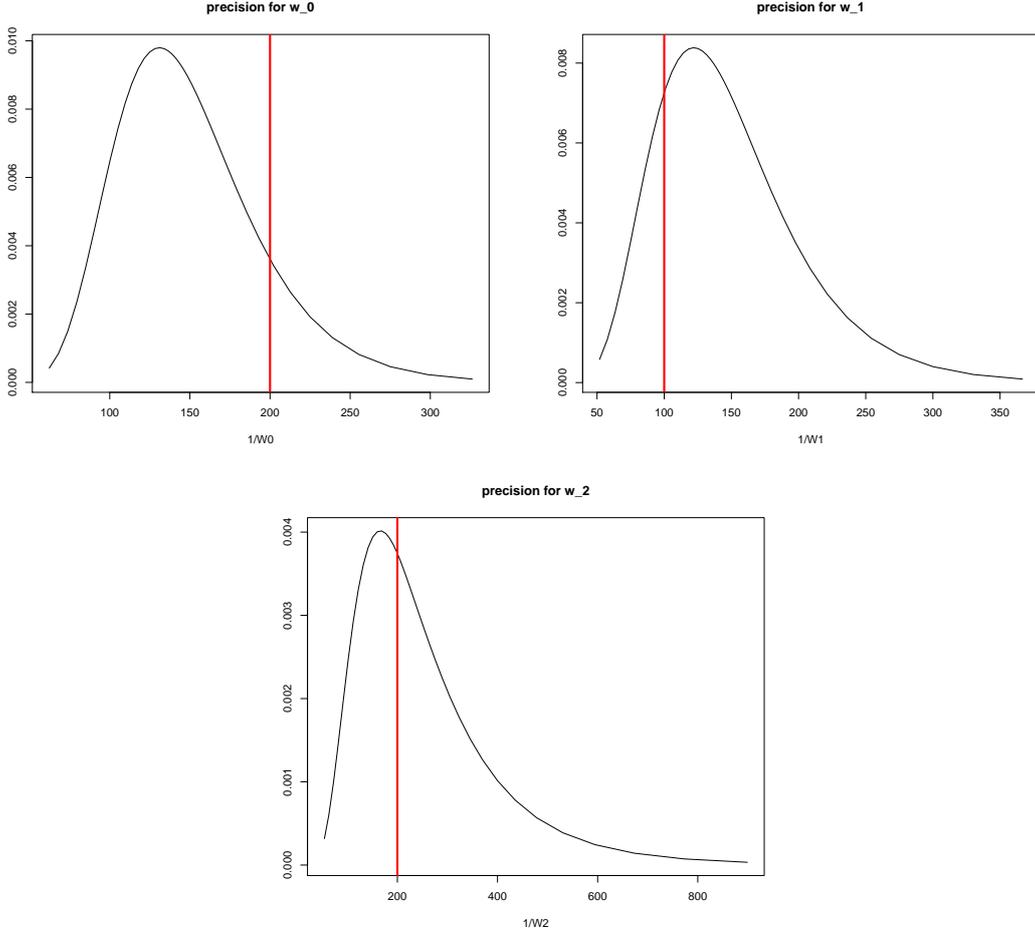


Figure 10: Posterior densities for the hyperparameters in the multiple Poisson dynamic regression example. Red lines indicate true simulated values.

### Example 5: A first order spatio-temporal dynamic model

In the following two examples, we simulate data from two versions of a non-stationary Gaussian spatio-temporal dynamic model without covariates (Vivar and Ferreira, 2009) in order to demonstrate how even complex dynamic models can be easily fitted using the INLA library. We begin with a non-stationary first-order Gaussian spatio-temporal dynamic model, where for each time  $t$  and area  $s$ ,  $t = 1, \dots, T$ ;  $s = 1, \dots, S$ , the response  $y_{ts}$  is specified as:

$$\mathbf{y}_t = \mathbf{F}'_t \mathbf{x}_t + \boldsymbol{\omega}_{1t}, \quad \boldsymbol{\omega}_{1t} \sim \text{PGMRF}(\mathbf{0}_s, \mathbf{W}_1^{-1}) \quad (18)$$

$$\mathbf{x}_t = \mathbf{G}_t \mathbf{x}_{t-1} + \boldsymbol{\omega}_{2t}, \quad \boldsymbol{\omega}_{2t} \sim \text{PGMRF}(\mathbf{0}_s, \mathbf{W}_2^{-1}) \quad (19)$$

where,  $\mathbf{y}_t = (y_{t1}, \dots, y_{tS})'$  denote the observed field at time  $t$ ,  $\mathbf{F}_t = \mathbf{I}_s$ ,  $\mathbf{G}_t = \rho \mathbf{I}_s$ ,  $\mathbf{0}_s$  is an  $S \times S$  null matrix, and  $\mathbf{I}_s$  is the  $S \times S$  identity matrix. The errors  $\boldsymbol{\omega}_{1t} = (\omega_{1t1}, \dots, \omega_{1tS})$  and  $\boldsymbol{\omega}_{2t} = (\omega_{2t1}, \dots, \omega_{2tS})$  are independent and modeled as proper Gaussian Markov random fields (PGMRF). Matrices  $\mathbf{W}_1^{-1}$  and  $\mathbf{W}_2^{-1}$  describe the spatial covariance structure of  $\boldsymbol{\omega}_{1t}$  and  $\boldsymbol{\omega}_{2t}$  respectively. We modeled precision matrices  $\mathbf{W}_j$ ,  $j = (1, 2)$ , as  $\mathbf{W}_j = \mathbf{D}_j \left( \mathbf{I}_s - \frac{\phi_j}{\lambda_{max}} \mathbf{C} \right)$ , with  $\mathbf{C}$  being a structure matrix defined as

$$C_{k,l} = \begin{cases} c_k & \text{if } k = l, \\ -h_{k,l} & \text{if } k \in d_l, \\ 0 & \text{otherwise,} \end{cases}$$

$d_l$  is the set of neighbors of area  $l$ ,  $h_{k,l} > 0$  is a measure of similarity between areas  $k$  and  $l$  (here we assume that  $h_{k,l} = 1$ ) and  $c_k = \sum_{l \in d_k} h_{k,l}$ .  $\lambda_{max}$  is the maximum eigenvalue of matrix  $\mathbf{C}$ ;  $\mathbf{D}_j = \tau_j \text{diag}(d_1, \dots, d_S)$ ,  $\tau_j$  are scale parameters and  $0 \leq \phi_j < 1$  control the degree of spatial correlation.

We simulated a time series of 30 times for each of the 100 areas of North Carolina's map (that is,  $S = 100$  and  $T = 30$ ). This map is available in R from `spdep` package (Bivand, 2010). Inference is performed for the state vector  $\mathbf{x}_t$  as well as for the scale and correlation parameters,  $\tau_1$ ,  $\tau_2$ ,  $\phi_1$ ,  $\phi_2$ , but not for  $\rho$ , whose value was fixed in one before analysis, leading to a non-stationary process.

For the implementation of this model using the INLA library, it is necessary to specify the precision matrices  $\mathbf{W}_1$  and  $\mathbf{W}_2$  through a generic model. This can be done using option `model='generic1'` in the formula to be called by the INLA library. This option requires that the structure matrix  $\mathbf{C}$  be passed as a file containing only the non-zero entries of the matrix. The file must contain three columns, where the first two ones contain the row and column indices of the non-zero entries of matrix  $\mathbf{C}$ , and the third column contain the corresponding non-zero values of structure matrix  $\mathbf{C}$ . The code in the appendix shows how this matrix can be built in R. For further details of how to specify structure matrices in INLA for use in the fitting of spatio-temporal models see Schrödle and Held (2009).

The comparison between simulated and predicted values of observations and states for some instant times can be seen in maps of Figures 11 and 12. Figure 13 also shows the series of simulated and predicted values for the states at area 20 and its neighbors. Predicted values closely followed the simulated series in all cases. Precision and correlation parameters were also well estimated even when default INLA values for hyperprior parameters and initial values were specified (see Figure 14 and Table 2).



Figure 11: Maps of simulated (left) and predicted values (right) for observations at times 2, 7 and 15. Gray and white areas represent positive and negative values respectively.



Figure 12: Maps of simulated (left) and predicted values (right) for  $X_1$  state vectors at times 2, 7 and 15. Gray and white areas represent positive and negative values respectively.

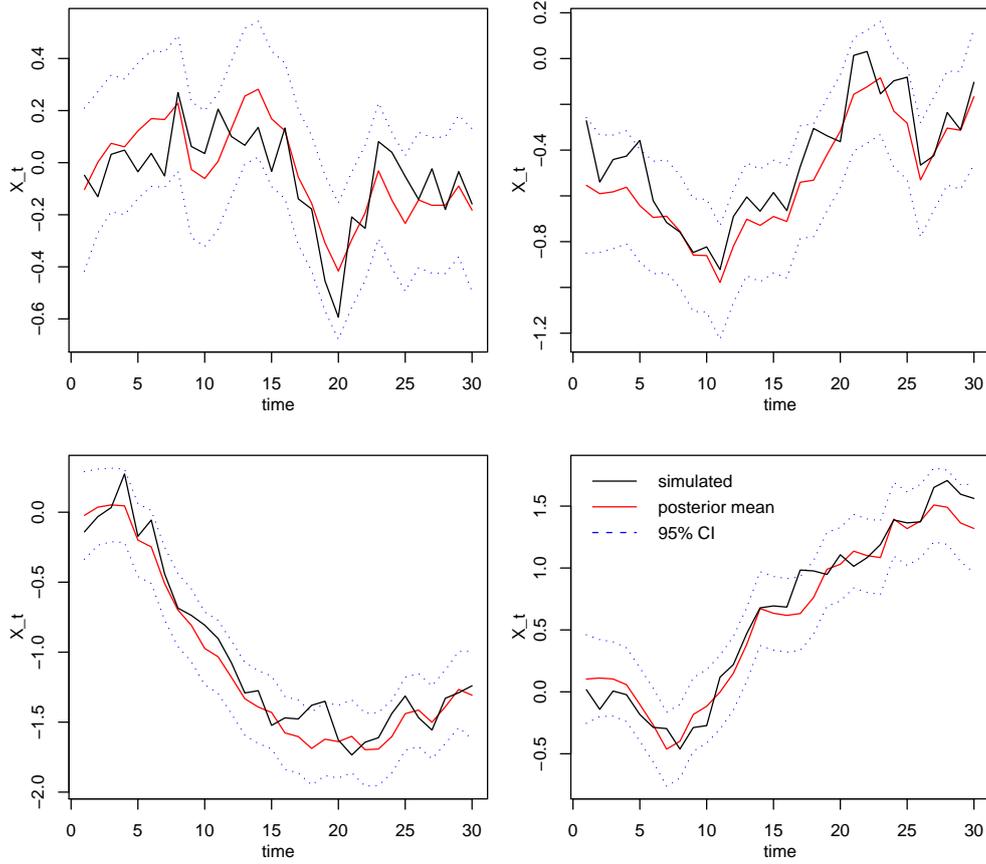


Figure 13: Simulated and predicted values (posterior mean and 95% credibility interval) for the states  $X_t$  in the 20th area (a) and in its neighbors (b-d) in the first order spatio-temporal dynamic model.

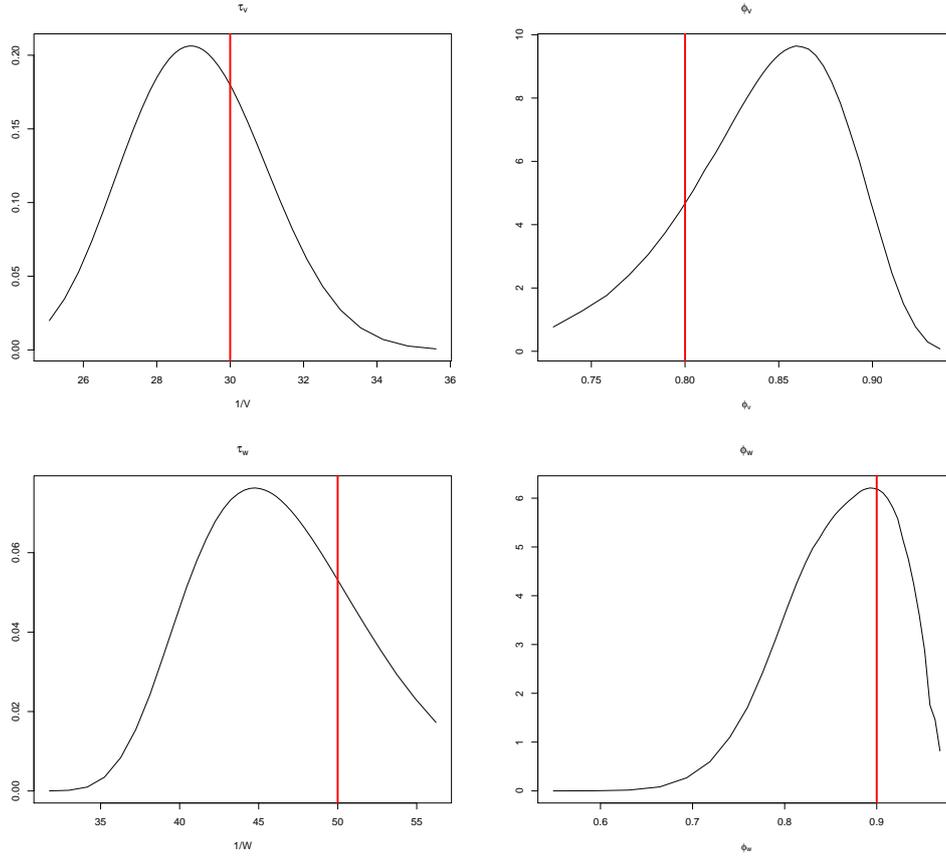


Figure 14: Posterior densities for precision (left) and correlation (right) hyperparameters in the first order spatio-temporal dynamic model. Red lines indicate true simulated values.

```

# summary of posterior estimates for hyperparameters (with default priors)
              mean          sd 0.025quant  0.5quant 0.975quant
\tau_1       29.1136768 1.86428401 25.6859129 29.0508193 32.9132280
\phi_1        0.8413540 0.04269526  0.7447260  0.8467934  0.9099569
\tau_2       46.1026702 4.84260536 37.5116052 45.8080660 55.6872788

\phi_2        0.8601783 0.06016452  0.7297083  0.8666207  0.9539419
# summary of posterior estimates for hyperparameters (with informative priors)
              mean          sd 0.025quant  0.5quant 0.975quant
\tau_1       28.6717312 1.99469330 25.0406355 28.6215894 32.6838160
\phi_1        0.8248531 0.05570660  0.6948307  0.8342335  0.9086119
\tau_2       47.9191173 6.46777489 37.4767242 47.1194431 62.0289891
\phi_2        0.8752391 0.06834365  0.7287361  0.8849223  0.9741523

# summary of posterior estimates for hyperparameters (with vague priors)
              mean          sd 0.025quant  0.5quant 0.975quant
\tau_1       29.0759014 1.84111689 25.6902778 29.0130694 32.8368471
\phi_1        0.8406674 0.04269136  0.7442875  0.8461058  0.9092824
\tau_2       45.9520470 4.73169579 37.5168869 45.6788380 55.2713507
\phi_2        0.8593318 0.06006310  0.7293980  0.8656297  0.9531633

```

Table 2. Summary of the posterior for the hyperparameters. True simulated values were:  $\tau_1 = 30$ ,  $\tau_2 = 50$ ,  $\phi_1 = 0.8$ , and  $\phi_2 = 0.9$ .

### Example 6: A second order spatio-temporal dynamic model

Now we will simulate data from a non-stationary second-order Gaussian spatio-temporal dynamic model without covariates (Vivar and Ferreira, 2009), which can be specified as:

$$\mathbf{y}_t = \mathbf{F}'_t \mathbf{x}_t + \boldsymbol{\omega}_{1t}, \quad \boldsymbol{\omega}_{1t} \sim \text{PGMRF}(\mathbf{0s}, \mathbf{W}_1^{-1}) \quad (20)$$

$$\mathbf{x}_t = \mathbf{G}_t \mathbf{x}_{t-1} + \boldsymbol{\omega}_{23t}, \quad \boldsymbol{\omega}_{23t} \sim \text{PGMRF}(\mathbf{0s}, \mathbf{W}_{23}^{-1}) \quad (21)$$

where,

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{x}_{1t} \\ \mathbf{x}_{2t} \end{pmatrix}, \quad \mathbf{F}_t = \begin{pmatrix} \mathbf{I}s \\ \mathbf{0s} \end{pmatrix}, \quad \mathbf{G}_t = \begin{pmatrix} \rho_1 \mathbf{I}s & \rho_1 \mathbf{I}s \\ \mathbf{0s} & \rho_2 \mathbf{I}s \end{pmatrix}, \quad \boldsymbol{\omega}_{23t} = \begin{pmatrix} \boldsymbol{\omega}_{2t} \\ \boldsymbol{\omega}_{3t} \end{pmatrix} \quad \text{and} \quad \mathbf{W}_{23}^{-1} = \begin{pmatrix} \mathbf{W}_2^{-1} & \mathbf{0s} \\ \mathbf{0s} & \mathbf{W}_3^{-1} \end{pmatrix},$$

and the rest of notation follows example 5.

Once again we simulate a time series of 30 times for each of the 100 areas of North Carolina's map. Inference is performed for the state vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  as well as for the scale and correlation parameters,  $\tau_j$ ,  $\phi_j$ ,  $j = (1, 2, 3)$ , but not for  $\rho_1$  and  $\rho_2$ , whose values were fixed in one before analysis, leading to a non-stationary process.

Appendix A shows some parts of the R code used to fit this model with INLA considering  $n = 100$  areas and  $k = 30$  times. Further details can be found in the R script accompanying this report.

As in example 5, for the implementation of this model in INLA it is necessary to specify the precision matrices  $\mathbf{W}_j$ , associated to the error vectors  $\boldsymbol{\omega}_{jt}$ , through a generic model using option `model='generic1'` in the formula to be called by the INLA library.

Note that precision parameter for observations in this and the above example is declared as fixed in the call to fit the model (first element of the list in `control.data`), because it is also specified through a generic model in the first line of the formula.

Initial values for the hyperparameters in this case must be carefully chosen, as this model is highly sensitive to these choices. Figures 15 and 16 show how a small change in one the initial values can change significantly the posterior densities of the hyperparameters. In this case we considered two sets of initial values as follows: `set1=( $\tau_v = 50, \tau_{\omega_1} = 100, \tau_{\omega_2} = 100, \phi_v = 0.8, \phi_{\omega_1} = 0.9, \phi_{\omega_2} = 0.9$ )` and `set2=( $\tau_v = 100, \tau_{\omega_1} = 100, \tau_{\omega_2} = 100, \phi_v = 0.8, \phi_{\omega_1} = 0.9, \phi_{\omega_2} = 0.9$ )`, where the only difference in the two sets is the initial value for the first hyperparameter, which change from 50 to 100. This small change impact not only that parameter, but also changes the posterior of the other hyperparameters.

	mean	sd	0.025quant	0.5quant	0.975quant
<code>\tau_1</code>	28.7823505	2.68308132	23.8774631	28.6585845	34.3797637
<code>\phi_1</code>	0.7789623	0.06329654	0.6416379	0.7842687	0.8864998
<code>\tau_2</code>	37.5158060	7.72405954	24.2701416	36.8375799	54.5635119
<code>\phi_2</code>	0.9434251	0.02341778	0.8854153	0.9483026	0.9751655
<code>\tau_3</code>	44.4007158	6.63377490	34.3423897	43.3199276	60.0260982
<code>\phi_3</code>	0.7850396	0.06270681	0.6524924	0.7887360	0.8942694

Table 1. Summary of the posterior for the hyperparameters. True simulated values were:  $\tau_v = 30, \tau_{\omega_1} = 50, \tau_{\omega_2} = 50, \phi_v = 0.8, \phi_{\omega_1} = 0.9$  and  $\phi_{\omega_2} = 0.9$ .

Maps of simulated and predicted values for observations and states in some areas are also shown in Figures 17 to 19.

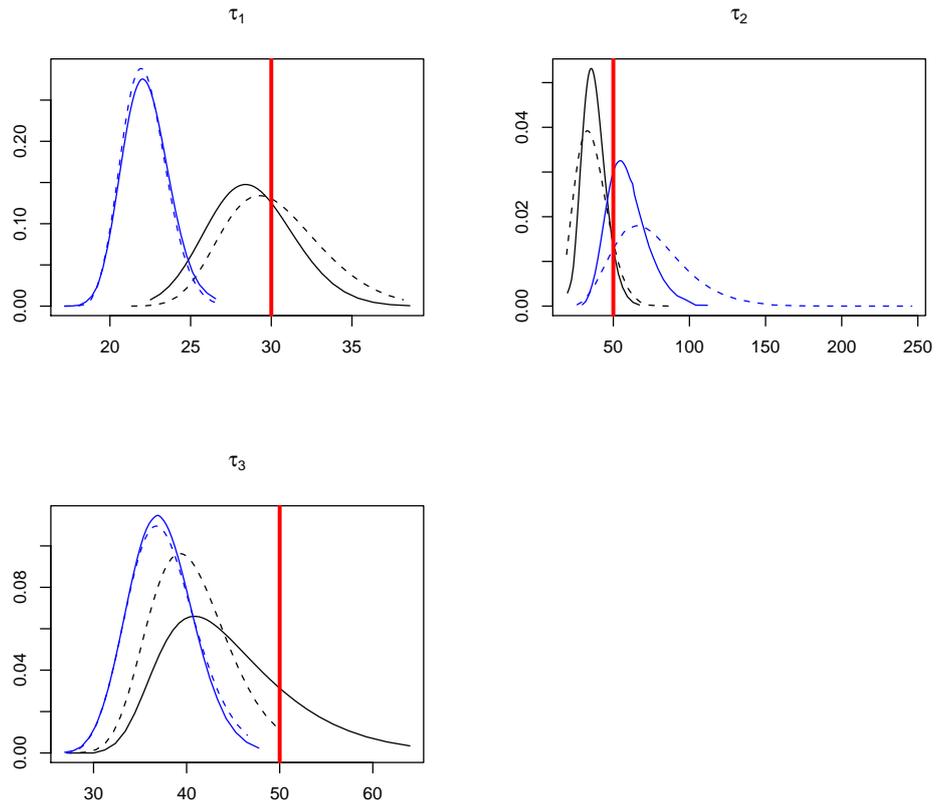


Figure 15: Posterior densities for the precision parameters in the 2nd order spatio-temporal dynamic model. Solid lines come from informative priors, whilst dotted lines come from vague priors. Lines in black are for the first set of initial values and lines in blue are for the second set of initial values (see text for details). True simulated values are indicated in red.

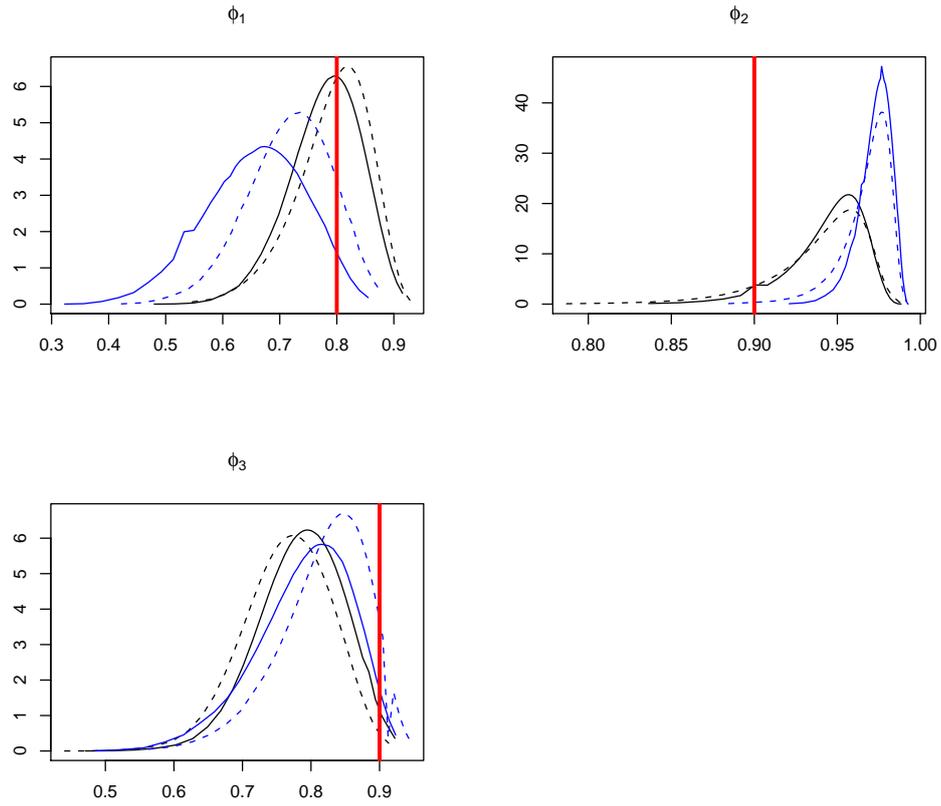


Figure 16: Posterior densities for the correlation parameters in the 2nd order spatio-temporal dynamic model. Solid lines come from informative priors, whilst dotted lines come from vague priors. Lines in black are for the first set of initial values and lines in blue are for the second set of initial values (see text for details). True simulated values are indicated in red.



Figure 17: Maps of simulated (left) and predicted values (right) for observations at times 2, 7 and 15. Gray and white areas represent positive and negative values respectively.



Figure 18: Maps of simulated (left) and predicted values (right) for  $X_1$  state vectors at times 2, 7 and 15. Gray and white areas represent positive and negative values respectively.



Figure 19: Maps of simulated (left) and predicted values (right) for  $X_2$  state vectors at times 2, 7 and 15. Gray and white areas represent positive and negative values respectively.

## 5 Case studies

In this section we use some worked examples from the literature to illustrate how a relevant data analysis with DLMS can be performed using INLA. The examples include dynamic models with Gaussian and Poisson observation densities, temporal trend and seasonality components as well as external covariates. When possible, comparison with results from the literature using other inference methods is provided. The full code to fit all models in this section is also provided in the R script accompanying this report.

### Example 7: UK Gas consumption

The first worked example to be analyzed corresponds to the quarterly UK gas consumption from 1960 to 1986, in millions of therms. Details on this dataset can be found in Durbin and Koopman (2001, p. 233). Following Dethlefsen and Lundbye-Christensen (2006) here we use the (base 10) logarithm of the UK gas consumption as response, which is assumed to be normal distributed and we fit a model with a first order polynomial trend ( $T_t$ ) with time-varying coefficients and an unstructured seasonal component ( $S_t$ ), also varying over time. Therefore, the observational and system equations are given by

$$\begin{aligned}
 y_t &= \log_{10}(UKgas)_t = T_t + S_t + \nu_t, & \nu_t &\sim N(0, V), & t &= 1, \dots, n \\
 T_t &= T_{t-1} + \beta_{t-1} + \omega_{1t}, & \omega_{1t} &\sim N(0, W_1), & t &= 2, \dots, n & (22) \\
 \beta_t &= \beta_{t-1} + \omega_{2t}, & \omega_{2t} &\sim N(0, W_2), & t &= 2, \dots, n & (23) \\
 S_t &= -(S_{t-1} + S_{t-2} + S_{t-3}) + \omega_{3t}, & \omega_{3t} &\sim N(0, W_3), & t &= 4, \dots, n & (24)
 \end{aligned}$$

Approximate inference in this case is performed using a mixed approach where the polynomial trend in system equation (22) is equated to zero and merged with the observational equation yielding an augmented model of dimension  $n + (n - 1)$  and two different likelihoods. The first  $n$  elements (actual observations) of this augmented structure are Gaussian distributed while the remaining  $n - 1$  elements are forced to be zero and considered as observed with a high and fixed precision. The slope term in equation (23), which follow a random walk evolution, was modeled using model option “`rw1`” from INLA library, whilst for the seasonal term in equation (24) we used model option “`seasonal`”. The main parts of the code to formulate and fit this model with INLA are shown next. The full code to fit this model using the INLA library is available from the R script accompanying this report.

The decomposition of the time series in trend, slope and seasonal components and its comparison with results obtained by Dethlefsen and Lundbye-Christensen (2006) with the `sspir` R package, which uses an extended Kalman filtering approach to inference, is shown in Figure 20. Results with the two approaches were very similar. The amplitude of the seasonal term remains virtually constant from 1960–1971, then it increases during the period 1971–1979 and finally it stabilizes again.

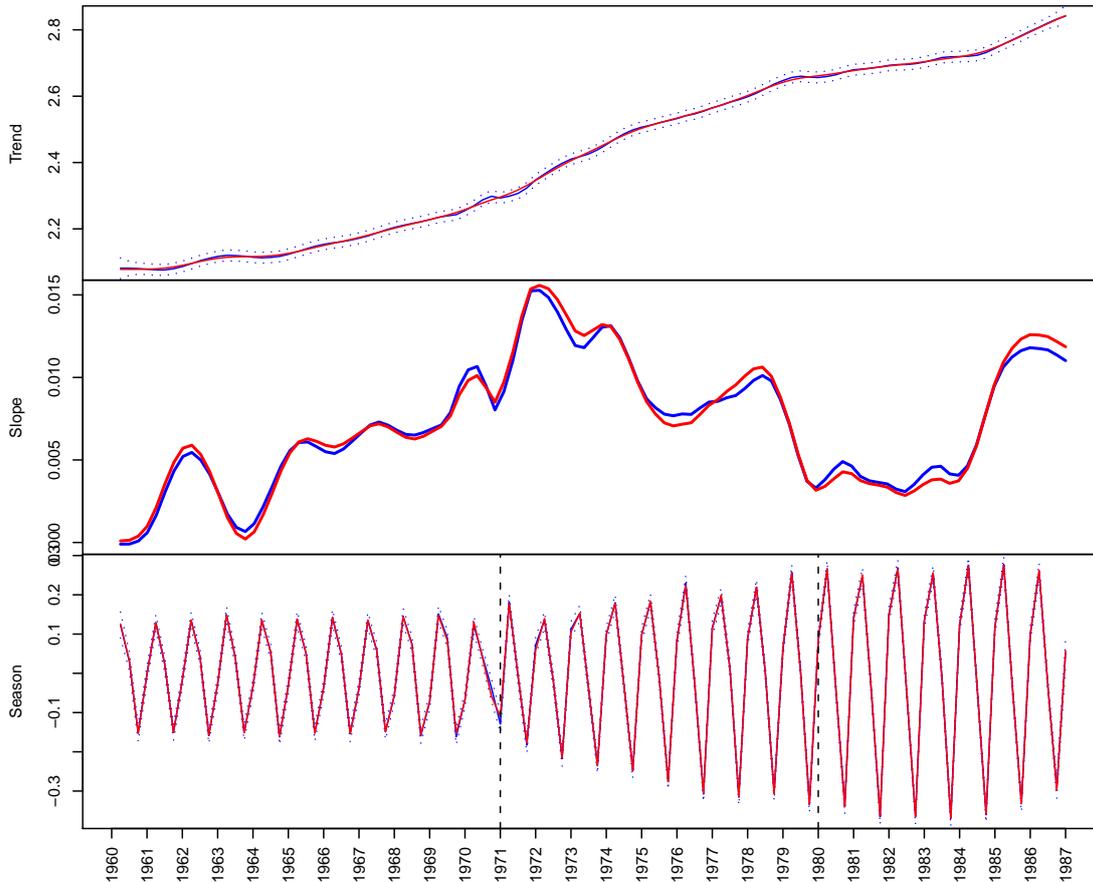


Figure 20: Time-varying trend, slope, and seasonal components in the UK gas consumption series obtained with INLA (blue lines) and with the sspir package (red lines). Dotted lines represent 95% credibility intervals for INLA estimates.

### Example 8: Van drivers

This is a classical example of a generalized linear dynamic model. Here the response  $y_t$  corresponds to the monthly numbers of light goods van drivers killed in road accidents in Great Britain, from January 1969 to December 1984 (192 observations). A seat belt law (intervention) was introduced on January 31st, 1983. The interest is in quantifying the effect of the seat belt legislation law on the number of deaths. For further information about the data set see Harvey and Durbin (1986) and Durbin and Koopman (2000).

This dataset has been previously analysed with INLA in a time series setting, assuming that the squared root of the counts  $y_t$  follows a Gaussian distribution. For details of this implementation see Martino and Rue (2010). Here we follow Dethlefsen and Lundbye-Christensen (2006) and use a generalized linear dynamic model for Poisson data with a 13-dimensional latent process, consisting of an intervention parameter, seat belt, changing value from zero to one in February 1983, a constant monthly seasonal term ( $S_t$ ), and a temporal trend ( $T_t$ ) modelled as

a random walk. The observational and system equations for this model are as follows

$$y_t \sim \text{Poisson}(\mu_t)$$

$$\log(\mu_t) = \lambda_t = T_t + \alpha * \text{seatbelt} + S_t, \quad t = 1, \dots, n \quad (25)$$

$$T_t = T_{t-1} + \omega_{1t}, \quad \omega_{1t} \sim N(0, W), \quad t = 2, \dots, n \quad (26)$$

$$S_t = -(S_{t-1} + \dots + S_{t-11}), \quad t = 12, \dots, n \quad (27)$$

The trend and seasonal terms in the linear predictor in (25) can be directly modeled using “rw1” and “seasonal” model options from INLA library as shown in the following code:

```
i <- j <- 1:n      # indices for T_t and S_t
ww <- rep(1,n)     # weights for T_t

formula <- y ~ belt + ww + f(i, ww, model="rw1", param=c(1,0.0005)) +
          f(j, model="seasonal", season.length=12) -1

r <- inla(formula, data = data.frame(belt,ww,i,j,y),
          family = "poisson", control.predictor=list(compute=TRUE))
```

Further coding details can be found in the R script accompanying this report. The estimated trend and the effect of the seat belt intervention as well as its comparison with results obtained with the `sspir` package (Dethlefsen and Lundbye-Christensen, 2006) can be displayed in Figure 21.

The posterior mean for  $\alpha$  parameter in Eq. (25), which represents the effect of the seat belt law on the number of deaths, was  $-0.283$ ; this corresponds to a reduction in the number of deaths of 24.63%. It agrees with the corresponding values reported by Durbin and Koopman (2000) and Dethlefsen and Lundbye-Christensen (2006) for this parameter, which were  $-0.280$  and  $-0.285$ , respectively.



Figure 21: Number of vandrivfers killed and estimated trend + intervention. Solid and dotted lines in blue correspond to the posterior mean and 95% credibility intervals, respectively, obtained with the INLA library. The line in red corresponds to the estimated trend + intervention with the sspir package.

### Example 9: Mumps

In this worked example the response  $y_t$  corresponds to the monthly registered cases of mumps in New York City from January 1928 to June 1972. This data set was previously studied by Hipel and McLeod (1994). According to Dethlefsen and Lundbye-Christensen (2006), the incidence of mumps are known to show seasonal behavior and a variation in trend during the study period. For this data set we use a generalized linear dynamic model for Poisson data. Following Dethlefsen and Lundbye-Christensen (2006) the mumps incidence was modelled here with a first order polynomial trend ( $T_t$ ) with time-varying coefficients and a time-varying harmonic seasonal component ( $H_t$ ). The observational and system equations for this model are as follows

$$\begin{aligned}
y_t &\sim \text{Poisson}(\mu_t) \\
\log(\mu_t) = \lambda_t &= T_t + H_t, & t = 1, \dots, n \\
T_t &= T_{t-1} + \beta_{t-1} + \omega_{1t}, & \omega_{1t} \sim N(0, W_1), & t = 2, \dots, n & (28) \\
\beta_t &= \beta_{t-1} + \omega_{2t}, & \omega_{2t} \sim N(0, W_2), & t = 2, \dots, n & (29) \\
H_t &= a_t \cos\left(\frac{2\pi}{12}t\right) + b_t \sin\left(\frac{2\pi}{12}t\right), & t = 1, \dots, n & (30) \\
a_t &= a_{t-1} + \omega_{3t}, & \omega_{3t} \sim N(0, W_3), & t = 2, \dots, n & (31) \\
b_t &= b_{t-1} + \omega_{4t}, & \omega_{4t} \sim N(0, W_4), & t = 2, \dots, n & (32)
\end{aligned}$$

As in example 7, in this case we use a mixed approach where the polynomial trend in system equation (28) is equated to zero and merged with the observational equation yielding an augmented model of dimension  $n + (n - 1)$  and two different likelihoods. The first  $n$  elements (actual observations) of this augmented structure are Poisson distributed while the remaining  $n - 1$  elements are forced to be zero and considered as observed with a high and fixed precision. The slope and seasonal terms in equations (29) to (32), which follow a random walk evolution, are modelled with model option “rw1”. The main parts of the code to formulate and fit this model with INLA are shown next. The full code is available from the R script accompanying this report.

```

# building the augmented model
# -----
m <- n-1
Y <- matrix(NA, n+m, 2)
Y[1:n, 1] <- mumps
Y[1:m + n, 2] <- 0

## indices for the INLA library
# -----
i <- c(1:n, 2:n) # indices for T_t
j <- c(rep(NA,n), 1:m) # indices for T_{t-1}
weight1 <- c(rep(NA,n), rep(-1,m)) # weights for T_{t-1}
l <- c(rep(NA,n), 1:m) # indices for \beta_{t-1}
weight2 <- c(rep(NA,n), rep(-1,m)) # weights for \beta_{t-1}
w1 <- c(rep(NA,n), 2:n) # indices for w_{1,t}
q <- c(1:n, rep(NA,m)) # indices for a_t
cosine <- c(cosw,rep(NA,m)) # weights for a_t
rr <- c(1:n, rep(NA,m)) # indices for b_t
sine <- c(sinw,rep(NA,m)) # weights for b_t

# formulating the model
# -----
formula <- Y ~ cosine + f(q, cosine, model="rw1",param=c(1,0.01),initial=4) +
  sine + f(rr, sine, model="rw1",param=c(1,0.01),initial=4) +
  f(l, weight2, model="rw1",param=c(1,0.2),initial=4) +
  f(i, model="iid", initial=-10, fixed=TRUE) +
  f(j, weight1, copy="i") +
  f(w1, model="iid") -1

```

```

# call to fit the model
# -----
r <- inla(formula, data = data.frame(cosine,sine,i,j,weight1,l,weight2,q,rr,w1),
         family = c("poisson","gaussian"),
         control.data = list(list(),list(initial=10, fixed=TRUE)),
         control.predictor=list(compute=TRUE))

```

It is important to note that in the formulation of this model, the seasonal terms, following an RW1 process, must be declared first in the formula to be passed to the inla function, followed by the terms in the equations that forms the augmented model. Otherwise the INLA library can made a wrong interpretation of the indices of these terms, which can lead to misleading results.

The comparison of the results obtained with the INLA library and the sspir package for the variation of mumps incidence are shown in Figure 22. Results were very similar for the two approaches. According to Figure 22, seasonal pattern of incidence changes slowly, as can be seen in the decreasing behavior of the peak-to-trough ratio and peak location series. The location of the incidence's peak also changes from middle/late April in the beginning of the study period to late May in the last four years.

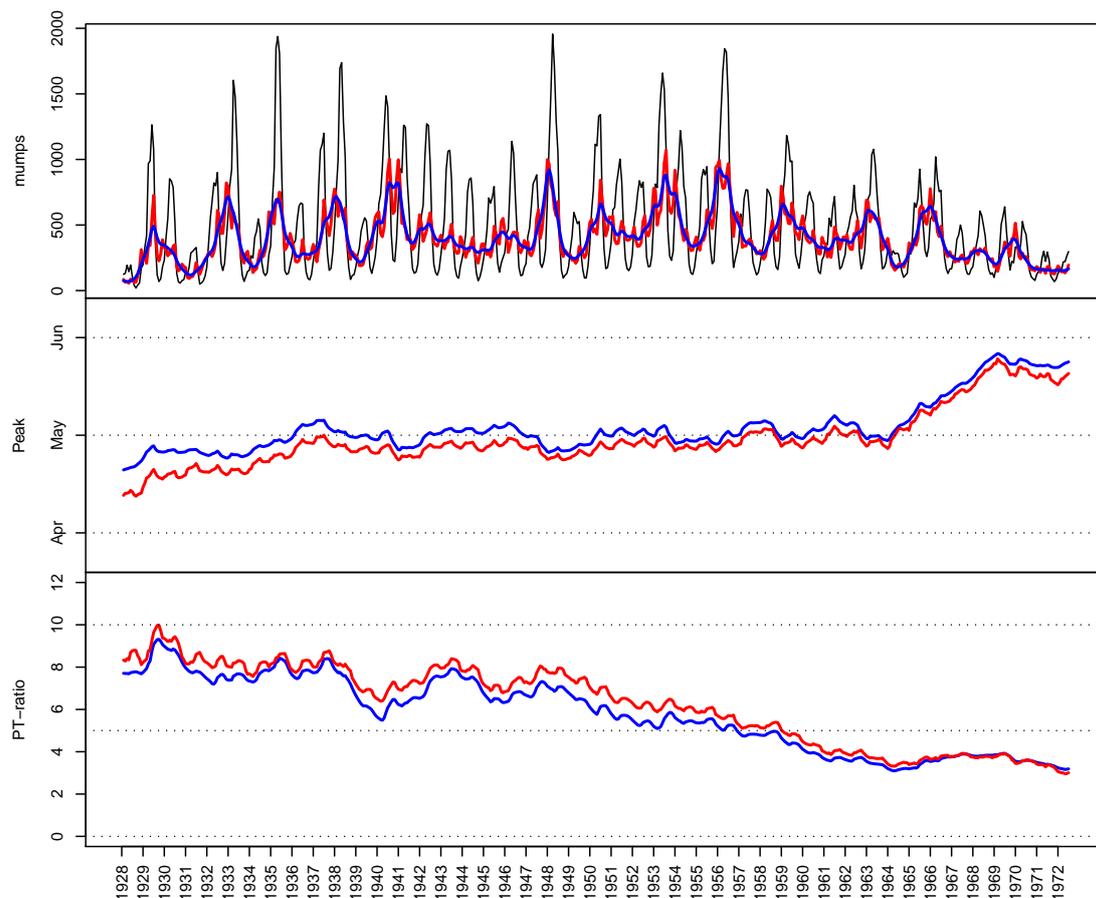


Figure 22: Comparison between INLA (red lines) and sspir (blue lines) results for the variation in the incidence of mumps in New York city from 1927 to 1972. The upper frame shows the observed number of cases with the de-seasonalized trend superimposed. The middle frame shows the location of the peak of the seasonal pattern. The lower frame shows the variation in the peak-to-trough ratio over the period.

### Example 10: Market share

In our last worked example we analyze percent market share for a consumer product. This example was fully analyzed in Pole et al. (1994, Chapter 5). The model for market share utilize weekly available information for 1990 and 1991 on product price and measures of promotional activity. The objectives are to determine a model with good predictive power and to assess the importance of suggested explanatory variables. The response  $y_t$  is assumed to be Gaussian distributed. For this data set we use a dynamic regression model with three covariates, **price**, **prom** and **cprom**, where **price** is the measured price relative to a number competitor's average prices; **prom** and **cprom** are producer and competitor promotion indices. Following Pole et al. (1994), the level is modelled as fixed and the regression coefficients have a random walk evolution. One point identified as outlier on week 34 of 1990 and excluded from the analysis in Pole et al. (1994), was also excluded in our analysis. The observational and system equations

for this model are as follows

$$y_t = \alpha_t + \beta_{1t}\text{price}_t + \beta_{2t}\text{prom}_t + \beta_{3t}\text{cprom}_t + \nu_t, \quad \nu_t \sim N(0, V), \quad t = 1, \dots, n$$

$$\beta_{1t} = \beta_{1,t-1} + \omega_{1t}, \quad \omega_{1t} \sim N(0, W_1), \quad t = 2, \dots, n \quad (33)$$

$$\beta_{2t} = \beta_{2,t-1} + \omega_{2t}, \quad \omega_{2t} \sim N(0, W_2), \quad t = 2, \dots, n \quad (34)$$

$$\beta_{3t} = \beta_{3,t-1} + \omega_{3t}, \quad \omega_{3t} \sim N(0, W_3), \quad t = 2, \dots, n \quad (35)$$

As in example 4, the simple random walk evolution of regression coefficients avoids the need for an augmented structure. Therefore, the model was formulated considering an “**rw1**” model for each regression coefficient. Figure 23 shows the predicted market share values and the estimated level obtained with the INLA library and that reported in Pole et al. (1994) using the BATS software.

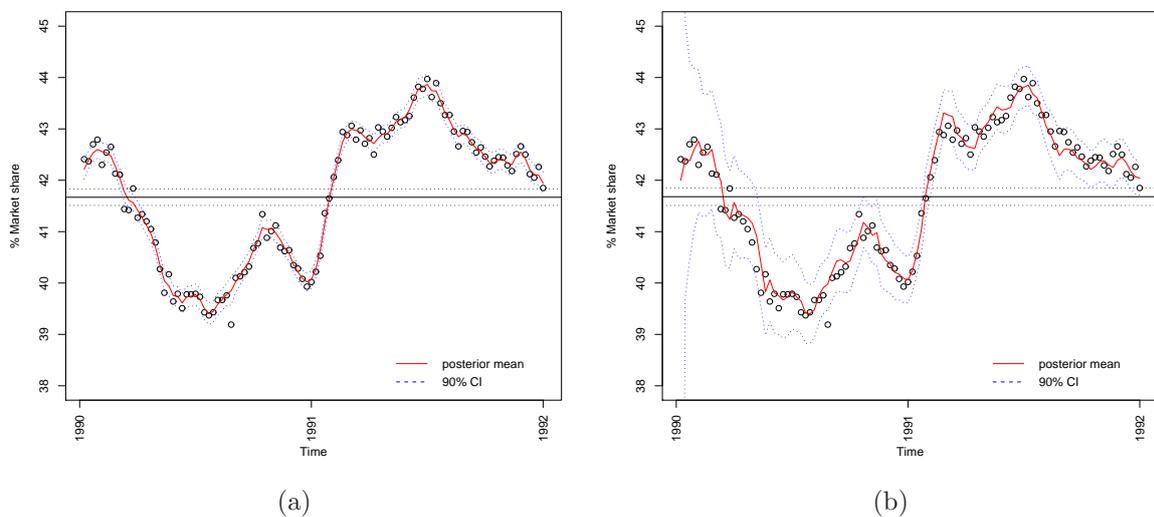


Figure 23: Observed and predicted values (posterior mean and 90% credibility interval) for the market share example using the INLA library (a) and the BATS software (b). Horizontal black lines in both plots indicate the estimated level with its 90% credibility interval.

Estimated regression coefficients for regressors are displayed in Figure 24. The promotion coefficient **prom** varies between 0 and 0.3, being nearly zero for almost the entire two year period. As pointed out by Pole et al. (1994), this suggests that the company’s promotional activities have little effect on market share.

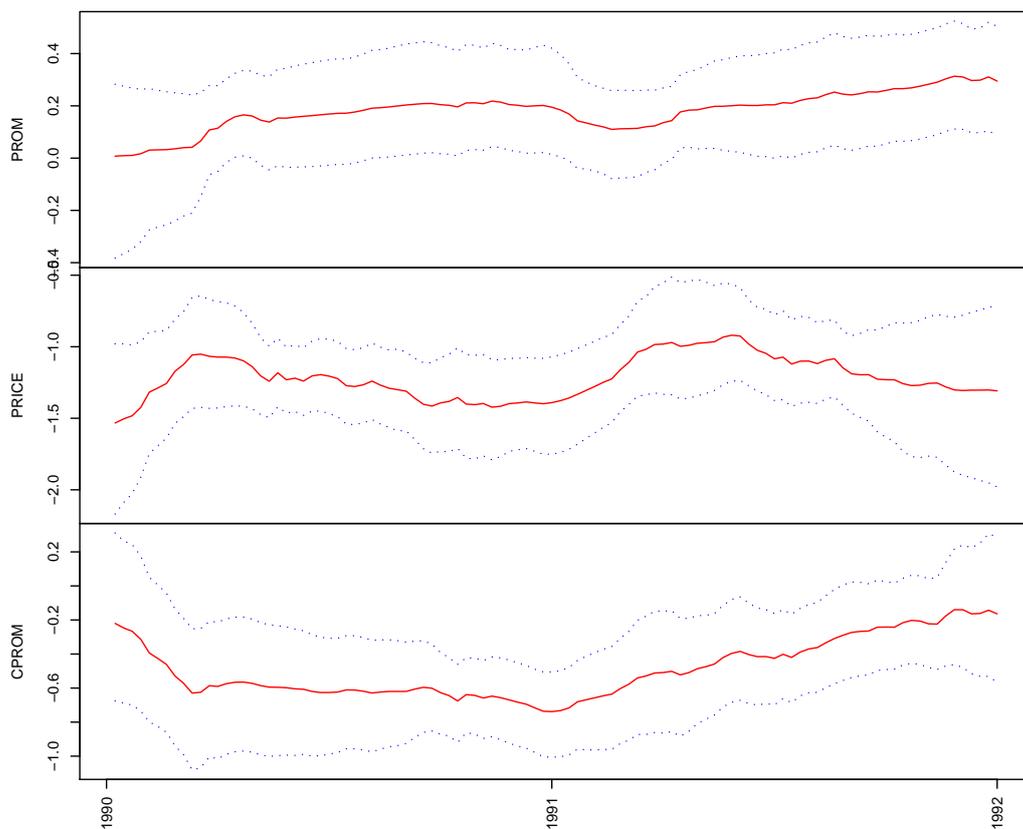


Figure 24: Estimated regression coefficients for the market share example. Solid red and dotted blue lines indicate posterior mean and 95% credibility interval, respectively.

Results of the week-by-week forecasts for the first five weeks of 1992 under four different scenarios, as considered in Pole et al. (1994), is shown in Table 1 using INLA and BATS software. The four scenarios were:

1. `prom` and `cprom` indices set to 0,
2. `prom` set to its first five values of 1990 and `cprom` set to zero,
3. `prom` set to zero and `cprom` set to its first five values of 1990,
4. `prom` and `cprom` set to their first five values of 1990.

Relative price was maintained fixed in all cases at 0.206, the final value for 1991.

Table 1: Forecasts for percent market share for the first four weeks of 1992 with INLA and BATS software

Week	INLA			BATS		
	mean Scenario1	0.05q	0.95q	mean Scenario1	0.05q	0.95q
1992/1	41.40	41.19	41.61	41.40	41.04	41.76
1992/2	41.40	41.18	41.61	41.40	41.04	41.76
1992/3	41.40	41.18	41.61	41.40	41.03	41.77
1992/4	41.40	41.18	41.62	41.40	41.03	41.77
1992/5	41.40	41.18	41.62	41.40	41.03	41.77
	Scenario2			Scenario2		
1992/1	41.36	41.14	41.59	41.36	41.00	41.73
1992/2	41.32	41.07	41.56	41.31	40.93	41.69
1992/3	41.27	41.00	41.54	41.25	40.86	41.65
1992/4	41.25	40.97	41.53	41.23	40.83	41.63
1992/5	41.22	40.92	41.52	41.19	40.77	41.61
	Scenario3			Scenario3		
1992/1	41.28	40.91	41.64	41.23	40.84	41.62
1992/2	41.27	40.90	41.66	41.22	40.82	41.62
1992/3	41.27	40.88	41.67	41.22	40.82	41.62
1992/4	41.27	40.87	41.68	41.22	40.81	41.63
1992/5	41.26	40.83	41.71	41.20	40.79	41.62
	Scenario4			Scenario4		
1992/1	41.24	40.86	41.63	41.19	40.79	41.59
1992/2	41.19	40.77	41.62	41.13	40.71	41.55
1992/3	41.14	40.69	41.62	41.07	40.63	41.51
1992/4	41.12	40.64	41.62	41.05	40.60	41.50
1992/5	41.08	40.55	41.63	41.00	40.53	41.47

## 6 Concluding remarks

In this report we propose a framework to perform approximate inference in linear and generalized linear dynamic models using the INLA library. We illustrate our approach through a series of simulated and worked examples ranging from simple univariate models to realistically complex space-time dynamic models. Our approach allows an easy specification of complex dynamic models in R using a formula language as is routinely done with the most common linear and generalized linear models. The proposed methodology outperforms current approaches in the literature of dynamic models in several respects:

- Unknown precision parameters and its credibility intervals are directly estimated with INLA jointly with the state parameters, unlike other approaches in the literature, which do not estimate the unknown variance parameters automatically. The `sspir` package (Dethlefsen and Lundbye-Christensen, 2006) for example requires the combination of numerical maximization algorithms with the output of the iterated extended Kalman smoother, while the BATS software (Pole et al., 1994) use a discount factor approach to model unknown variances. The `SsfPack` package (Koopman et al., 1999) provides punctual estimates of the hyperparameters of state space models, but it requires further Monte Carlo simulation in order to get the confidence intervals through some bootstrap procedure as proposed for example in Franco et al. (2008).
- Our approach is able to deal with spatio-temporal observations in an easy way, as shown in examples 5 and 6. To the best of our knowledge there are no other computational approaches in the literature currently available to deal with this kind of data in a framework of dynamic models.
- Missing values in the covariates are allowed.

Reasonable results were found for most of the examples considered in this report using the default INLA values for the hyperprior parameters and for the initial values of these parameters. However, for some models this choice can greatly impact the final results. Therefore, a sensitivity analysis to the choice of that values is highly recommended.

The extension of the proposed framework to consider multivariate observations is straightforward. The approach has also potential to be applied/extended to other classes of models such as models with errors in covariates. This is subject of current research.

## References

- Bivand, R. (2010) `spdep`: Spatial dependence: weighting schemes, statistics and models. R package version 0.5-9. <http://CRAN.R-project.org/package=spdep>
- Dethlefsen, C. and Lundbye-Christensen, S. (2006) Formulating State Space Models in R with Focus on Longitudinal Regression Models. *Journal of Statistical Software*, **16**, 1–15.
- Durbin J, Koopman S.J. (2000) Time Series Analysis of Non-Gaussian Observations Based on State Space Models from both Classical and Bayesian Perspectives (with discussion). *Journal of the Royal Statistical Society Series B*, **62**, 3–56.
- Durbin, J. and Koopman, S.J. (2001) Time Series Analysis by State Space Methods. Oxford University Press.
- Ferreira, M. A. R. and De Oliveira, V. (2007) Bayesian reference analysis for Gaussian Markov Random Fields. *Journal of Multivariate Analysis*, **98**, 789–812.
- Franco, G.C., Santos, T.R., Ribeiro, J.A. and Cruz, F.R.B. (2008) Confidence intervals for hyperparameters in structural models. *Communications in Statistics: Simulation and Computation*, **37**, 486–497.
- Gamerman, D. (1998) Markov Chain Monte Carlo for Dynamic Generalised Linear Models. *Biometrika*, **85**, 215–227.

- Godsill, S., Doucet, A. and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, **99**, 156–168.
- Gordon, N.J., Salmond, D.J. and Smith, A.F.M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings F*, **140**, 107–113.
- Hipel, K.W. and McLeod, I.A. (1994) Time Series Modeling of Water Resources and Environmental Systems. Elsevier Science Publishers B.V. (North-Holland).
- Harvey, A.C. and Durbin, J. (1986) The Effects of Seat Belt Legislation on British Road Casualties: A Case Study in Structural Time Series Modelling (with discussion). *Journal of the Royal Statistical Society series A*, **149**, 187–227.
- Helske, J. (2010) KFAS: Kalman filter and smoothers for exponential family state space models. R package version 0.6.0. <http://CRAN.R-project.org/package=KFAS>.
- Jazwinski, A.H. (1970) *Stochastic Processes and filtering Theory*. San Diego: Academic Press.
- Koopman, S.J., Shephard, N. and Doornik, J.A. (1999) Statistical algorithms for models in state space using SsfPack 2.2. *Econometrics Journal*, **2**, 113–166.
- Martino, S. and Rue, H. (2010) Implementing Approximate Bayesian Inference using Integrated Nested Laplace Approximation: a manual for the inla program. Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. Compiled on April 8, 2010. Available at: <http://www.math.ntnu.no/~hrue/inla/manual.pdf>.
- Migon, H.S., Gamerman, D., Lopes, H.F. and Ferreira, M.A.R. (2005) Dynamic Models. In: *Handbook of Statistics*, **25**, D.K. Dey and C.R. Rao (Eds), Elsevier:North-Holland, 553–588.
- Pole, A., West, M. and Harrison, J. (1994) Applied Bayesian Forecasting and Time Series Analysis, New York, Chapman & Hall.
- Pole, A. and West, M. (1990) Efficient Bayesian learning in non-linear dynamic models. *Journal of Forecasting*, **9**, 119–136.
- Petris, G. (2010) dlm: Bayesian and Likelihood Analysis of Dynamic Linear Models. R package version 1.1-1. <http://CRAN.R-project.org/package=dlm>.
- R Development Core Team (2010) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Reis, E.A., Salazar, E. and Gamerman, D. (2006) Comparison of Sampling Schemes for Dynamic Linear Models. *International Statistical Review*, **74**, 203–214.
- Ripley, B.D. (2002) Time Series in R 1.5.0. *R News*, **2**, 2–7.
- Rue, H. and Follestad, T. (2002) GMRFLib: a C-library for fast and exact simulation of Gaussian Markov random fields. Preprint series in statistics No 1/2002, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. Available at: <http://www.stat.ntnu.no/preprint/statistics/2002/S1-2002.ps>.
- Rue, H. and Held, L. (2005) Gaussian Markov Random Fields: Theory and Applications. London: Chapman and Hall/CRC Press.
- Rue H. and Martino S. (2007) Approximate Bayesian Inference for Hierarchical Gaussian Markov Random Fields Models. *Journal of Statistical Planning and Inference*, **137**, 3177–3192.

Rue, H., Martino, S. and Chopin, N. (2009) Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society series B*, **71**, 319–392.

Schrödle, B. and Held, L. (2009) Evaluation of case reporting data from Switzerland: Spatio-temporal disease mapping using INLA. Technical Report, Biostatistics Unit, Institute of Social and Preventive Medicine, University of Zurich.

Vivar, J. C. and Ferreira, M. A. R. (2009) Spatiotemporal models for gaussian areal data. *Journal of Computational and Graphical Statistics*, **18**, 658–674.

West, M., Harrison, P.J. and Pole, A. (1988) Bats - A user guide - Bayesian Analysis of Time Series - release 1.3 - June 1988, University of Warwick.

West, M. and Harrison, J. (1997) *Bayesian forecasting and dynamic models*. Second edition, New York:Springer.

## A R script for fitting the second order dynamic spatio-temporal model in example 5

```
## simulating the data set

# Loading North Carolina's map (it has 100 areas)
require(spdep)
ncfile <- system.file("etc/shapes/sids.shp", package="spdep")[1]
nc <- readShapePoly(ncfile)

# building the structure matrix (C)
nc.nb <- poly2nb(nc)
d <- sapply(nc.nb, length) # vector with number of neighbors
C <- diag(d) - nb2mat(nc.nb, style="B") # structure matrix

n <- length(d)

# simulated values for tau_i and phi_i (i=1,2,3)
tau <- c(30, 50, 50)
phi <- c(0.8, 0.9, 0.9)

# building the precision matrix
lamb.max <- max(eigen(C, only.values=TRUE)$values) # maximum eigenvalue of C matrix
Q1 <- (diag(n)-phi[1]/lamb.max*C)
Q2 <- (diag(n)-phi[2]/lamb.max*C)
Q3 <- (diag(n)-phi[3]/lamb.max*C)

myrmvnorm <- function(n, mu, S)
  sweep(matrix(rnorm(n*nrow(S))), n)%%chol(S), 2, mu)

# defining the length of time series (number of years)
k <- 30

set.seed(1)
```

```

# simulating observational and innovation errors
w1 <- t(myrmvnorm(k, rep(0,n), solve(tau[1]*Q1)))
w2 <- t(myrmvnorm(k, rep(0,n), solve(tau[2]*Q2)))
w3 <- t(myrmvnorm(k, rep(0,n), solve(tau[3]*Q3)))

# generating the time series for observations and states
yy <- x1 <- x2 <- matrix(0, n, k)
x1[,1] <- w2[,1]
x2[,1] <- w3[,1]
for (i in 2:k) {
  x2[,i] <- x2[,i-1] + w3[,i]
  x1[,i] <- x1[,i-1] + x2[,i-1] + w2[,i]
}
yy <- x1 + w1

### defining the Cmatrix to use with model='generic1' for w1, w2 and w3
st.cmat <- kronecker(C, diag(k))
c.mat <- list(i=unlist(apply(st.cmat!=0, 1, which)),
             j=rep(1:nrow(st.cmat), rowSums(st.cmat!=0)),
             Cij=st.cmat[st.cmat!=0])

### building the augmented model
### -----
nd <- n*k
Y <- matrix(NA, nd*3-2*n, 3)
Y[1:nd, 1] <- as.vector(t(yy))
Y[1:(nd-n) + nd, 2] <- 0
Y[1:(nd-n) + 2*nd-n, 3] <- 0

### indices for the f() function
### -----
id1 <- (1:nd)[-((1:n)*k)]
id2 <- (1:nd)[-c(1,((1:(n-1))*k)+1)]
ix1 <- c(1:nd, id2, rep(NA,nd-n)) ## indices for x1_t
ix1b <- c(rep(NA,nd), id1, rep(NA,nd-n)) ## indices for x1_{t-1}
wx1b <- c(rep(NA,nd), rep(-1,nd-n), rep(NA,nd-n)) ## weights for x1_{t-1}
ix2 <- c(rep(NA,nd),rep(NA,nd-n), id2) ## indices for x2_t
ix2b <- c(rep(NA,nd), rep(id1, 2)) ## indices for x2_{t-1}
wx2b <- c(rep(NA,nd), rep(-1,2*(nd-n))) ## weights for x2_{t-1}
iw1 <- c(1:nd, rep(NA,2*(nd-n))) ## indices for w1_t
iw2 <- c(rep(NA,nd), id2, rep(NA,nd-n)) ## indices for w2_t
iw3 <- c(rep(NA,nd),rep(NA,nd-n), id2) ## indices for w3_t

## formulating the model
## -----
# with default prior for precision parameters and initial \phi=0.5
formula1 <- Y ~ f(iw1, model="generic1", Cmatrix=c.mat, initial=c(log(50),0.5)) +
  f(ix1, model="iid", initial=-10, fixed=T) +
  f(ix1b, wx1b, copy="ix1") +
  f(ix2, model="iid", initial=-10, fixed=T) +

```

```

f(ix2b, wx2b, copy="ix2") +
f(iw2, model="generic1", Cmatrix=c.mat, initial=c(log(100),0.5)) +
f(iw3, model="generic1", Cmatrix=c.mat, initial=c(log(100),0.5)) -1

## call to fit the model
## -----
require(INLA)
r1 <- inla(formula1, data = data.frame(ix1,ix1b,wx1b,ix2,ix2b,wx2b,iw1,iw2,iw3),
  family = rep("gaussian",3),
  control.inla = list(h=0.1),
  control.data = list(list(initial=10, fixed=T),
  list(initial=10, fixed=T), list(initial=10, fixed=T)),
  control.predictor=list(compute=TRUE))

```