Simplified Integrated Nested Laplace Approximation

Simon N. Wood
School of Mathematics, University of Bristol, U.K.
simon.wood@bristol.ac.uk

January 28, 2019

Abstract

Integrated Nested Laplace Approximation provides accurate and efficient approximations for marginal distributions in latent Gaussian random field models. Computational feasibility of the original Rue et al. (2009) methods relies on efficient approximation of Laplace approximations for the marginal distributions of the coefficients of the latent field, conditional on the data and hyperparameters. The computational efficiency of these approximations depends on the Gaussian field having a Markov structure. This note provides equivalent efficiency without requiring the Markov property, which allows for straightforward use of latent Gaussian fields without a sparse structure, such as reduced rank multi-dimensional smoothing splines. The method avoids the approximation for conditional modes used in Rue et al. (2009), and uses a log determinant approximation based on a simple quasi-Newton update. The latter has a desirable property not shared by the most commonly used variant of the original method.

1 Introduction

Consider a regression model in which a response n-vector, $\mathbf{y}$, depends on covariates, x_j, via latent Gaussian random fields. For example, x_j might be spatial location, and y depends on a Gaussian Markov random field defined over space, or on a thin plate spline or Gaussian process model of spatial location. Or x_j might be a univariate covariate and the response depends on a cubic spline of x_j or on a latent Gaussian auto-regressive process indexed by x_j. Realizations of these latent fields can be written in terms of basis expansions $f(x_j) = \sum_k \beta_k b_k(x_j)$, where the β_k are coefficients and the $b_k(x_j)$ known functions. The prior for the random field is a Gaussian density on the coefficients, with hyper-parameters, $\boldsymbol{\theta}$. Spline based generalized additive models and Gaussian process models are familiar examples. Inference with such models can be based on empirical Bayes methods, discussed in Wood (2017), on stochastic simulation, exemplified by Umlauf et al. (2015), or on integrated nested Laplace approximation, also known as INLA (Rue et al., 2009; Sørbye and Rue, 2011; Martins et al., 2013; Rue et al., 2017). The latter offers a particularly efficient approach to full Bayesian inference, but the published methods rely heavily on sparse bases and sparse prior precision matrices. Stochastic simulation methods also

require sparsity to achieve computational efficiency. As discussed in Wood (2017) and elsewhere, in many circumstances the latent field priors imply quite strong smoothness, making a reduced rank dense basis expansion highly attractive computationally. This note therefore proposes a simple alternative to the key efficiency promoting approximations in Rue et al. (2009), which does not require sparsity, and possesses a quite attractive theoretical property. The result is an efficient method for fully Bayesian inference with the full range of smooth regression models discussed, for example, in Wood (2017), that can still be used in the sparse setting.

INLA obtains the marginal distributions of the elements of the p-vector of model coefficients, $\boldsymbol{\beta}$, and hyper parameter vector, $\boldsymbol{\theta}$, from

$$\pi(\beta_i \mid \mathbf{y}) = \int \pi(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})\pi(\boldsymbol{\theta} \mid \mathbf{y})d\boldsymbol{\theta} \text{ and } \pi(\theta_i \mid \mathbf{y}) = \int \pi(\boldsymbol{\theta} \mid \mathbf{y})d\boldsymbol{\theta}_{-i} \tag{1}$$

where a subscript '$-i$' denotes a vector without its ith element. Laplace approximations are used for the distributions in the integrands, and the integrals are evaluated numerically, either over a relatively coarse $\boldsymbol{\theta}$ grid, or using the approach described in section 6.5 of Rue et al. (2009) based on central composite designs computed by the algorithm in Sanchez and Sanchez (2005). In practice, the integration might also be skipped and $\boldsymbol{\theta}$ simply set to its posterior mode.

A first order Laplace approximation is used for the posterior of $\boldsymbol{\theta}$

$$\tilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \frac{\pi(\hat{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})}{\pi_G(\hat{\boldsymbol{\beta}} \mid \mathbf{y}, \boldsymbol{\theta})}$$

where $\hat{\boldsymbol{\beta}}$ is the maximizer of $\pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\theta})$ and $\pi_G(\boldsymbol{\beta} \mid \boldsymbol{\theta}, \mathbf{y}) = N(\hat{\boldsymbol{\beta}}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \mathbf{H}^{-1}$ and $\mathbf{H}$ is the Hessian of $-\log \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\beta}$ at $\hat{\boldsymbol{\beta}}$. Since π_G is evaluated at its mode the approximation is simply $\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\hat{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})/|\mathbf{H}|^{1/2}$. Usually $\mathbf{H}$ depends on $\boldsymbol{\theta}$, albeit slowly.

The key step in INLA, computationally and conceptually, is the approximation

$$\tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y}) \propto \frac{\pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})}{\pi_{GG}(\tilde{\boldsymbol{\beta}}_{-i} \mid \beta_i, \mathbf{y}, \boldsymbol{\theta})}, \tag{2}$$

where $\tilde{\boldsymbol{\beta}}$ maximizes $\pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\theta})$ given the constraint $\tilde{\beta}_i = \beta_i$, and π_{GG} is a Gaussian approximation to the density of $\boldsymbol{\beta}_{-i} \mid \beta_i, \mathbf{y}, \boldsymbol{\theta}$. We can of course approximate $\pi(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$ directly from $\pi_G(\boldsymbol{\beta} \mid \boldsymbol{\theta}, \mathbf{y})$, but this involves evaluating a Gaussian approximation well away from its mode, where it will often be inaccurate. In contrast, (2) only requires the evaluation of a Gaussian approximation at its mode, and is therefore more accurate. Furthermore, at worst, a relative error in π_{GG} at its mode translates into an equivalent relative error in approximation (2), which should be compared with the behaviour of the marginal based on π_G, where the error simply grows as we move into the tails. Finally, the approximate $\pi(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$ is always re-normalized in practice, which eliminates any component of the approximation error due to inaccuracies in π_{GG} at its mode which are β_i-independent. These are the key insights underpinning Rue et al. (2009).

If we base π_{GG} directly on the mode and Hessian of $\log \pi(\boldsymbol{\beta}_{-i} \mid \beta_i, \mathbf{y}, \boldsymbol{\theta})$ then (2) is exactly the Laplace approximation to $\int \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\theta})d\boldsymbol{\beta}_{-i}$, immediately providing access to formal results on approximation accuracy as discussed in Shun and McCullagh (1995) and Rue et al. (2009).

Unfortunately direct evaluation of the required Hessian is computationally prohibitive when it has to be performed for each β_i. Much cheaper is to base π_{GG} on the conditional density implied by π_G, in which case the Hessian is constant and

$$\tilde{\boldsymbol{\beta}}_{-i} = \hat{\boldsymbol{\beta}}_{-i} + \boldsymbol{\Sigma}_{-i,i}\Sigma_{i,i}^{-1}(\beta_i - \hat{\beta}_i), \tag{3}$$

leading to the approximation $\pi(\beta_i \mid \boldsymbol{\theta}, \mathbf{y}) \propto \pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})$, which is demonstrably a substantial improvement on directly using the marginal from π_G. Rue et al. (2009) use (3), but achieve slightly better approximation performance by also approximating the dependence on β_i of the Hessian of $\log \pi(\boldsymbol{\beta}_{-i} \mid \beta_i, \mathbf{y}, \boldsymbol{\theta})$. They offer two alternatives. The first exploits the heuristic that only elements of $\boldsymbol{\beta}_{-i}$ showing sufficiently high correlation to β_i according to π_G need be considered when approximating how the Hessian varies with β_i: this leads to efficient computation for Markov models, but appears difficult to exploit in the non-Markov case. The second, recommended, approach replaces the log determinant of the required Hessian with a first order Taylor approximation about $\hat{\boldsymbol{\beta}}$. The required log determinant derivative is computationally costly for non-Markov models.

The proposal here is to employ modified approximations that do not increase the leading order cost in the sparse Markov case, but are also efficient in the dense non-Markov case. To use INLA at all we have to compute $\hat{\boldsymbol{\beta}}$ using Newton's method, which in turn requires the Hessian $\mathbf{H}$ and its Cholesky factorization $\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{H}$. The Hessian with respect to $\boldsymbol{\beta}_{-i}$ at $\hat{\boldsymbol{\beta}}$ is $\mathbf{H}_{-i,-i}$, and its Cholesky factor can be computed directly from $\mathbf{R}$ at $O(p^2)$ cost in the dense case. The update starts from $\mathbf{R}_{\bullet,-i}$ and zeroes the elements on its sub-diagonal by applying Givens rotations from the left, as detailed in the appendix. Routine `choldrop` in R package `mgcv` (Wood, 2017) will do this. If $\mathbf{R}$ is a sparse matrix, routine `cholmod_updown` from the `suitesparse` library (Davis, 2006) will achieve the same. The proposed method modification then has two parts.

1. Use the numerically exact $\tilde{\boldsymbol{\beta}}_{-i}$ in place of (3). Several steps of Newton's method, with fixed Hessian $\mathbf{H}_{-i,-i}$, can be used to find $\tilde{\boldsymbol{\beta}}_{-i}$, starting from (3). Each Newton step requires gradient evaluation, at $O(np)$ cost in the dense case, and less in the sparse case, plus two triangular solves using the Cholesky factor of $\mathbf{H}_{-i,-i}$, at $O(p^2)$ cost in the dense case, and the cost of the sparse triangular solves otherwise. Newton iteration convergence is slowed, but still guaranteed, with a fixed Hessian (see Wood, 2015, §5.1.1, for example).

2. Approximate the required Hessian of $\log \pi(\boldsymbol{\beta}_{-i} \mid \beta_i, \mathbf{y}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\beta}_{-i}$ by an appropriate BFGS update of $\mathbf{H}_{-i,-i}$ (see Nocedal and Wright, 2006, §5.6.1 for example). With correct structuring of the update, the computational cost is that of gradient evaluation and triangular solves involving the Cholesky factor of $\mathbf{H}_{-i,-i}$. It is shown below that this update has the desirable property of giving a determinant bounded between that of $\mathbf{H}_{-i,-i}$ and the true target Hessian determinant. This property is not shared by the first order Taylor approximation of the log determinant.

Part 1 is simply removing an approximation, and is therefore an accuracy improvement on Rue et al. (2009). Part 2 will likely be less accurate than a first order Taylor expansion of the log determinant when β_i is very close to $\hat{\beta}_i$ but, by virtue of Theorem 1 below, will always become more accurate at some point as β_i moves away from $\hat{\beta}_i$, unless the log determinant is really linear in β_i.

2 The method

Here is the complete algorithm for computing a single marginal density approximation $\tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$. Let $\boldsymbol{\mathcal{D}}_{2j}$ denote the $2j \times 2j$ diagonal matrix with leading diagonal $-1, 1, -1, 1 \ldots$, and $\mathbf{u}_0$ be a zero column matrix. Let $\tilde{\boldsymbol{\beta}}(\beta_i)$ denote the posterior mode given β_i at a fixed value, and let $\boldsymbol{\Delta}_j^i$ denote $\boldsymbol{\Delta}_j$ with an extra zero inserted at element i.

1. Compute the Cholesky factor of $\tilde{\mathbf{H}}_0 = \mathbf{H}_{-i,-i}$ by update of the Cholesky factor of $\mathbf{H}$.
2. For each β_i in a grid of evaluation points repeat steps 3–7.
3. Use Newton's method with fixed Hessian $\tilde{\mathbf{H}}_0$ to find $\tilde{\boldsymbol{\beta}}_{-i}(\beta_i)$, starting from (3).
4. Compute a set of J steps $\{\boldsymbol{\Delta}_j\}$.
5. For $j = 0, \ldots, J-1$
   Compute $\mathbf{h} = \tilde{\mathbf{H}}_0 \boldsymbol{\Delta}_j + \mathbf{u}_j \boldsymbol{\mathcal{D}}_{2j} \mathbf{u}_j^\mathsf{T} \boldsymbol{\Delta}_j$ and $\mathbf{g} = \nabla_\beta \log \pi\{\tilde{\boldsymbol{\beta}}(\beta_i) + \boldsymbol{\Delta}_j^i, \mathbf{y}, \boldsymbol{\theta}\}$.
   Compute the matrix $\mathbf{u}_{j+1} = \left\{\mathbf{h}(\boldsymbol{\Delta}_j^\mathsf{T}\mathbf{h})^{-1/2}, \mathbf{g}_{-i}(\boldsymbol{\Delta}_j^\mathsf{T}\mathbf{g}_{-i})^{-1/2}, \mathbf{u}_j\right\}$.
6. Compute the determinant approximation $|\tilde{\mathbf{H}}_1| = |\tilde{\mathbf{H}}_0||\mathbf{I}_{2J} + \mathbf{u}_J^\mathsf{T} \tilde{\mathbf{H}}_0^{-1} \mathbf{u}_J \boldsymbol{\mathcal{D}}_{2J}|$.
7. Compute $\tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y}) = \pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta}) / |\tilde{\mathbf{H}}_1|^{1/2}$.
8. Re-normalize $\tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$.

In the work reported here $J = 1$ and $\boldsymbol{\Delta}_0/h = \hat{\boldsymbol{\beta}}_{-i} - \tilde{\boldsymbol{\beta}}_{-i}(\beta_i)$ for some small h. Step 5 implements the BFGS update given explicitly in Theorem 1, below. Theorem 1 is also the reason that the update is based on a small step from $\tilde{\boldsymbol{\beta}}_{-i}(\beta_i)$, rather than the whole step from $\hat{\boldsymbol{\beta}}_{-i}(\beta_i)$ to $\tilde{\boldsymbol{\beta}}_{-i}(\beta_i)$. Step 6 uses results from section 18.1 of Harville (1997). An alternative to steps 5 and 6 would be to directly update the Cholesky factor of $\tilde{\mathbf{H}}_0$ according to the BFGS update, however this would lose sparsity in the Markov case, while the given update works equally well in the sparse or dense cases. In practice all computations are with log determinants and densities, and any computation involving $\tilde{\mathbf{H}}_0^{-1}$ is accomplished via two triangular solves with the Cholesky factor of $\tilde{\mathbf{H}}_0$. Further computational savings are available by computing $\log \tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$ only on a relatively coarse grid of β_i values and using spline interpolation for evaluation and normalization. In the section 5 motorcycle example 16 β_i values were used, but further efficiency can be obtained by following Rue et al. (2009) and assuming a skew normal or skew t posterior, the parameters of which can then be obtained using far fewer evaluations.

At step 6, $|\tilde{\mathbf{H}}_0|$ is constant for a given i, which, given step 8, means that it does not actually have to be computed. This opens the possibility of avoiding the computation of the Cholesky factor of $\mathbf{H}_0$, and using an alternative computation of terms of the form $\mathbf{H}_0^{-1}\mathbf{x}$, where $\mathbf{x}$ is a vector. Specifically, if $\boldsymbol{\Sigma} = \mathbf{H}^{-1}$ then from basic properties of multivariate Gaussian densities $(\mathbf{H}_{-i,-i})^{-1}\mathbf{x} = (\boldsymbol{\Sigma}_{-i,-i} - \boldsymbol{\Sigma}_{-i,i}\Sigma_{i,i}^{-1}\boldsymbol{\Sigma}_{i,-i})\mathbf{x}$. Hence letting $\mathbf{x}_i^0$ denote $\mathbf{x}$ with an extra zero inserted at position i, $\mathbf{e}_i$ denote the ith column of the identity matrix, $\mathbf{d} = (\mathbf{H}^{-1}\mathbf{e}_i)_{-i}$ and $\delta = (\mathbf{H}^{-1}\mathbf{e}_i)_i$, we obtain $(\mathbf{H}_{-i,-i})^{-1}\mathbf{x} = (\mathbf{H}^{-1}\mathbf{x}_i^0)_{-i} - \mathbf{d}\mathbf{d}^\mathsf{T}\mathbf{x}/\delta$. This doubles the number of triangular solves required to compute each $(\mathbf{H}_{-i,-i})^{-1}\mathbf{x}$ term in both the Newton update and determinant correction, but enables the use of sparse matrix libraries lacking a suitable Cholesky update routine.

When using sparse model representations the β_i are usually directly interpretable as the quantities of interest, but in the reduced rank dense case this may not be true, and other linear combinations of coefficients may be of more interest. For example, the coefficients of a spline

smoother might be less interesting than the values of the evaluated spline at some values of its covariate. This is readily handled by defining an invertible linear transformation from $\boldsymbol{\beta}$ to transformed parameters of interest $\boldsymbol{\beta}' = \mathbf{A}\boldsymbol{\beta}$. If there are fewer than p identifiable parameters of interest it is always possible to augment them in order to create an invertible $\mathbf{A}$. Then the transformation can be applied to $\mathbf{H}$, its inverse and $\hat{\boldsymbol{\beta}}$, and the algorithm applied for $\boldsymbol{\beta}'$. To evaluate $\pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})$ and its gradient simply requires the inverse transform from $\boldsymbol{\beta}'$ to $\boldsymbol{\beta}$ and linear transformation of the gradient vectors.

3 Properties of the determinant update

The following theorem guarantees that in the small update step limit the determinant of the updated Hessian is bounded between the determinant of the Hessian at $\hat{\boldsymbol{\beta}}$ and the determinant of the true Hessian. This is not a property of all such low rank updates. For example, the symmetric-rank-1 update does not have this property, despite converging more rapidly to the true Hessian than BFGS, and even within the Broyden class of updates, only those 'between' the BFGS and Davidon-Fletcher-Powell updates have the property. The linear log determinant approximation used in the simplified Laplace approximation of Rue et al. (2009, §3.2.3) does not share this property.

Theorem 1. *Let $\tilde{\mathbf{H}}_0$ and $\tilde{\mathbf{H}}$ be respectively the initial Hessian and true Hessian with respect to $\boldsymbol{\beta}_{-i}$ at $\tilde{\boldsymbol{\beta}}(\beta_i)$, and assume that $\log \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\lambda})$ is regular with bounded third derivative. Let $\tilde{\mathbf{H}}_1$ denote the BFGS update of $\tilde{\mathbf{H}}_0$ based on a step $h\boldsymbol{\Delta}$ from $\tilde{\boldsymbol{\beta}}$ where $\|\boldsymbol{\Delta}\| = 1$. Then $|\tilde{\mathbf{H}}_1| \in [|\tilde{\mathbf{H}}_0| + O(h), |\tilde{\mathbf{H}}| + O(h)]$.*

Proof. The update is in the subspace of $\boldsymbol{\beta}_{-i}$, and the gradient vector of $\log \pi$ with respect to $\boldsymbol{\beta}_{-i}$ is zero at $\tilde{\boldsymbol{\beta}}$. Defining $f(\boldsymbol{\beta}_{-i}) = \log \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\lambda})$ where β_i is fixed on the right hand side, the BFGS update is

$$\tilde{\mathbf{H}}_1 = \tilde{\mathbf{H}}_0 + \frac{\tilde{\mathbf{H}}_0 \boldsymbol{\Delta}\boldsymbol{\Delta}^\mathsf{T} \tilde{\mathbf{H}}_0}{\boldsymbol{\Delta}^\mathsf{T} \tilde{\mathbf{H}}_0 \boldsymbol{\Delta}} + \frac{\nabla f(\tilde{\boldsymbol{\beta}}_{-i} + h\boldsymbol{\Delta}) \nabla f(\tilde{\boldsymbol{\beta}}_{-i} + h\boldsymbol{\Delta})^\mathsf{T}}{h\boldsymbol{\Delta}^\mathsf{T} \nabla f(\tilde{\boldsymbol{\beta}}_{-i} + h\boldsymbol{\Delta})}.$$

Now $\nabla f(\tilde{\boldsymbol{\beta}}_{-i} + h\boldsymbol{\Delta}) = \nabla^2 f(\tilde{\boldsymbol{\beta}}_{-i})\boldsymbol{\Delta} h + O(h^2)$, so $\tilde{\mathbf{H}}_1 = \tilde{\mathbf{H}}_2 + O(h)$ where

$$\tilde{\mathbf{H}}_2 = \tilde{\mathbf{H}}_0 + \frac{\tilde{\mathbf{H}}_0 \boldsymbol{\Delta}\boldsymbol{\Delta}^\mathsf{T} \tilde{\mathbf{H}}_0}{\boldsymbol{\Delta}^\mathsf{T} \tilde{\mathbf{H}}_0 \boldsymbol{\Delta}} + \frac{(\nabla^2 f(\tilde{\boldsymbol{\beta}}_{-i})\boldsymbol{\Delta})(\nabla^2 f(\tilde{\boldsymbol{\beta}}_{-i})\boldsymbol{\Delta})^\mathsf{T}}{\boldsymbol{\Delta}^\mathsf{T} \nabla^2 f(\tilde{\boldsymbol{\beta}}_{-i})\boldsymbol{\Delta}}.$$

Recalling that $\nabla^2 f(\tilde{\boldsymbol{\beta}}_{-i}) = \tilde{\mathbf{H}}$, this last is the BFGS update for the quadratic function $f(\tilde{\boldsymbol{\beta}}_{-i} + \boldsymbol{\Delta}) = f(\tilde{\boldsymbol{\beta}}_{-i}) + \boldsymbol{\Delta}^\mathsf{T} \tilde{\mathbf{H}} \boldsymbol{\Delta}/2$. In consequence we can apply Theorem 8.3 of Nocedal and Wright (2006), which directly implies that $|\tilde{\mathbf{H}}_2^{-1}||\tilde{\mathbf{H}}| \in [1, |\tilde{\mathbf{H}}_0||\tilde{\mathbf{H}}|]$. Hence $|\tilde{\mathbf{H}}_2| \in [|\tilde{\mathbf{H}}_0|, |\tilde{\mathbf{H}}|]$ and $|\tilde{\mathbf{H}}_1| \in [|\tilde{\mathbf{H}}_0| + O(h), |\tilde{\mathbf{H}}| + O(h)]$. □

4 Hyper-parameters

The hyper-parameters, $\boldsymbol{\theta}$, are often of less direct interest than $\boldsymbol{\beta}$, but when their marginal distributions are also required, the dense low-rank approach offers a computationally cheap alternative to the direct integration in (1) or the integration free approximation of Martins et al. (2013).

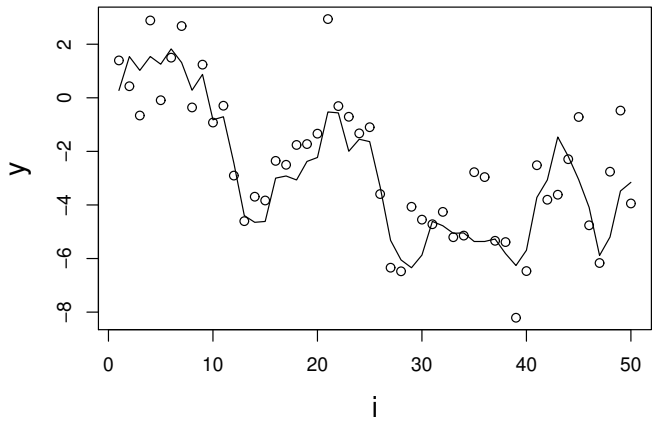

Figure 1: Example simulation for the first model considered in section 5, based on the first example from Rue et al. (2009). Open circles are data, y_i while the line joins the underlying f_i.

If the posterior modes, $\hat{\boldsymbol{\theta}}$, are estimated by Laplace approximate marginal likelihood maximization, as described in Wood et al. (2016) and Wood (2017), then we automatically have access to a Gaussian approximation for the posterior distribution of $\boldsymbol{\theta}$ based on $\hat{\boldsymbol{\theta}}$ and the Hessian of the log marginal likelihood plus any log prior on $\boldsymbol{\theta}$. This in turn provides a Gaussian approximation for $\pi(\boldsymbol{\theta}_{-i} \mid \theta_i)$. The approximations are most accurate with a little care in the choice of parameterization: for example if σ^2 is a variance component then a $\log \sigma$ parameterization is usually better than using σ directly. Re-using the ideas employed for $\tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$ we then have the approximation

$$\tilde{\pi}(\theta_i \mid \mathbf{y}) \propto \frac{\pi(\boldsymbol{\beta}^*, \boldsymbol{\theta}^*, \mathbf{y})}{\pi_G(\boldsymbol{\beta}^* \mid \boldsymbol{\theta}^*, \mathbf{y}) \pi_G(\boldsymbol{\theta}_{-i} \mid \theta_i)}$$

where $\boldsymbol{\beta}^*$ and $\boldsymbol{\theta}^*$ maximize $\pi(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{y})$ given a fixed value of θ_i. In fact $\pi(\boldsymbol{\beta}^*, \boldsymbol{\theta}^*, \mathbf{y})/\pi_G(\boldsymbol{\beta}^* \mid \boldsymbol{\theta}^*)$ is simply the maximized Laplace approximate marginal likelihood, so $\tilde{\pi}(\theta_i \mid \mathbf{y})$ can be readily evaluated given the method for Laplace approximate marginal likelihood maximization already used, provided only that this admits fixing of θ_i while optimizing $\boldsymbol{\theta}_{-i}$. Again it is straightforward to normalize $\tilde{\pi}(\theta_i \mid \mathbf{y})$.

5 Examples

As an illustrative comparison, the first example from Rue et al. (2009) was repeated to compare the full and simplified approximations proposed there to the modified method proposed here, and to the simple Gaussian approximation, π_G. The model from section 5.1 of Rue et al. is $y_i - f_i \sim t_3$ where $f_1 - \mu \sim N(0, 1)$, $f_i - \mu \sim N\{\phi(f_{i-1} - \mu), \sigma^2\}$ if $i = 2, \ldots, 50$, $\phi = 0.85$, $\sigma = 1$ and $\mu \sim N(0, 1)$. 1000 replicate data sets were simulated from the model, of which figure 1 shows one example. For each f_i, for each replicate, each approximation was compared to the results of Gibbs sampling using the JAGS package (Plummer, 2003, 2014). The Gibbs

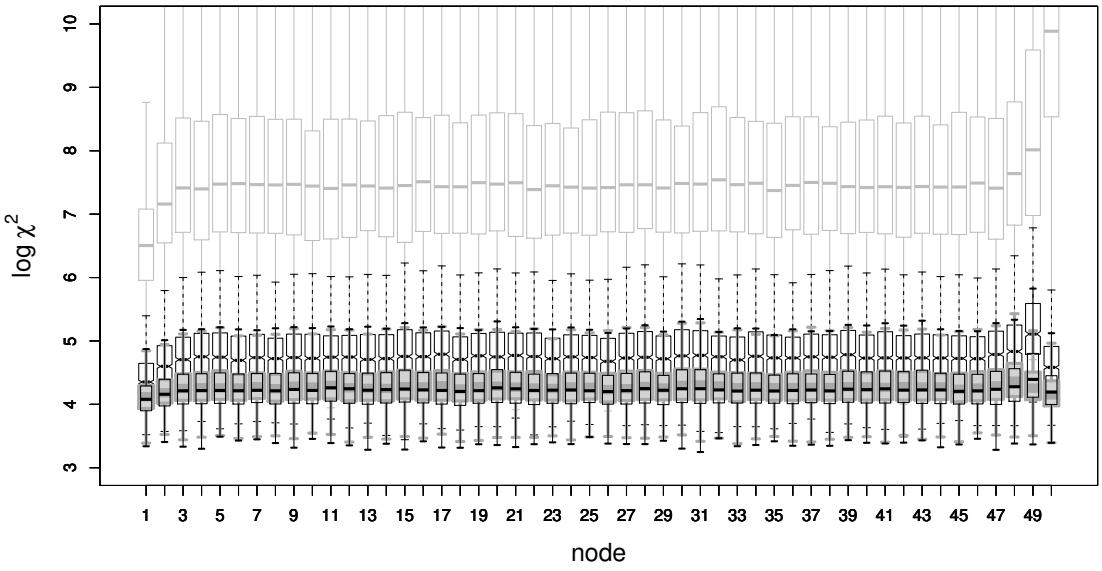

Figure 2: Superimposed boxplots showing the distribution of mismatch between four alternative approximations to each $\pi(f_i \mid y)$ and the truth, as represented by a long Gibbs sampling run, for the example illustrated in Figure1. For each of 1000 replicates, mismatch is measured by a $\log \chi^2$ statistic, so lower values indicate a better approximation. See section 5 for more detail. The horizontal axis gives the index i of each node, f_i, of the model. The light grey boxplots, with suppressed whiskers and tending to have larger $\log \chi^2$ values, are for the simple Gaussian approximation, $\pi_G(f_i \mid y)$. The notched box plots, with dashed whiskers and intermediate $\log \chi^2$ values, are for the simplified Laplace approximation recommended by Rue et al. (2009). The narrow black outlined boxplots are for the more expensive approximation from Rue et al. The wide boxplots with grey infill, 'behind' the thin black boxplots, are for the simple method proposed in section 2. The section 2 method and the expensive Rue et al. approximation show similar behaviour, substantially better than the alternatives.

sampling runs were of length 100000, thinned to 10000 to be approximately uncorrelated. For each f_i the simulated values were put in 50 equal width bins and a χ^2 statistic was computed for testing whether the simulations were generated by the marginal distribution as computed by each approximation. R code and JAGS code for the simulations is provided in the supplementary material. Figure 2 summarizes the results, giving boxplots for the log χ^2 statistics for each method by node, f_i. Outlier plotting has been suppressed for plotting clarity. The outliers show no obvious differences between the methods. For comparison, the log of the mean χ^2 statistics for each method are: 4.51, new method; 4.80, Rue et al. expensive approximation; 5.31, Rue et al. simplified Laplace; 21.38, Gaussian. Simply using the correct mode and a constant Hessian gives a log mean χ^2 of 4.89. The log expected χ^2 statistic for the true distribution would be 3.91. The same ordering applies for means of the log χ^2 statistic, and for the median, except that for the median the advantage of the new method over the more expensive INLA approximation is reversed, 4.26 to 4.22. All the INLA based approximations are much better than the simple Gaussian approximation. The method proposed here is competitive with the more expensive approximation proposed in Rue et al. (2009) and in this case is markedly better than the simplified version, although it should be noted that the log determinant derivative is zero for the simplified method, for this model.

As a second illustration consider the much overworked motorcycle crash test data from

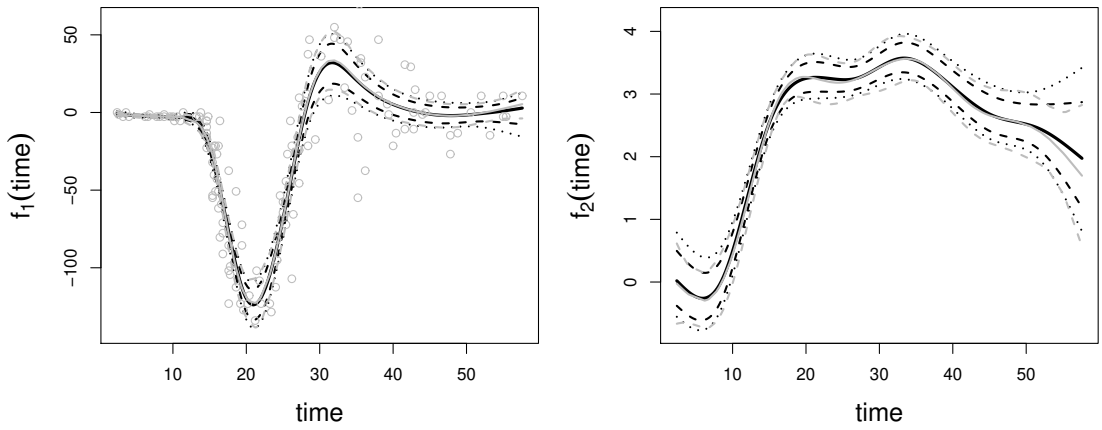

Figure 3: Credible intervals for the motorcycle crash data. The left panel shows the data as grey circles and the credible intervals for the mean. The 2.5%, 10%, 50%, 90% and 97.5% quantiles computed using the INLA method proposed here are shown as black dotted, dashed, solid, dashed and dotted curves, respectively. The mean and 95% credible limits from the simple Gaussian approximation are shown in grey. The right panel shows the same for the smooth function modelling the log standard deviation of the data.

Silverman (1985), in which the acceleration of the heads of crash test dummies was measured against time in simulated motorcycle accidents. A possible model for these data is

$$a_i \sim N\{f_1(t_i), e^{2f_2(t_i)}\}$$

where f_1 and f_2 are smooth functions allowing the mean and variance of acceleration, a_i, to vary smoothly with time, t_i. f_1 was represented using a rank 20 adaptive spline and f_2 a rank 10 thin plate regression spline (Wood, 2017). f_1 had 5 smoothing parameters, and f_2 had one. Wood et al. (2016) provide efficient empirical Bayesian methods for such models, but rely on the simple Gaussian approximation, π_G, for the posterior distribution of the model coefficients. However, given the Wood et al. (2016) methods, the INLA method proposed here is easily applied.

Figure 3 compares credible intervals generated using the INLA variant method proposed here and the simple Gaussian approximation given directly by the methods in Wood et al. (2016). The INLA intervals were computed for spline function values at evenly spaced times, using the linear transform method discussed in section 2: these values were then spline interpolated. The first integral in (1) was performed using the central composite design strategy from Rue et al. (2009, §6.5), with the outer design points placed on a contour of equal probability according to the Gaussian approximation to $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, automatically available when using the Wood et al. (2016) methods. The Gaussian and INLA intervals show quite marked differences for the log standard deviation, especially at the interval ends, and there are some differences even for the mean acceleration.

6 Discussion

The key to efficiency of INLA is the approximation of $\pi(\beta_i \mid \boldsymbol{\theta}, \mathbf{y})$, and the method for this proposed here has the advantages of simplicity; applicability in the case of dense and sparse model representations; the theoretically re-assuring property given by Theorem 1; competitive statistical performance with the more expensive approximation from Rue et al. (2009) and improved performance relative to the simplified Laplace approximation usually employed in practice. On the other hand it could be argued that Theorem 1 merely provides the minimum for a reasonable approximation, offering only the loosest of bounds on the approximation error for the determinant. A counter-argument is that the simplified Laplace approximation usually employed in INLA computations does not even satisfy these loose bounds, generally being more accurate in the centre of the distribution and less accurate in the tails: but improving tail behaviour is one of the main motivations for taking a fully Bayesian approach. In any case the section 2 method can always be improved by increasing J from 1 and choosing several update step. One possibility is to take a number of orthogonal steps forming a reduced rank basis for the space of relevant coefficients identified by the more expensive approximation in Rue et al. (2009).

The method proposed here is available in R package `mgcv` from version 1.8-27

Acknowledgement

I am grateful to a referee for pointing out that the explicit update of R can be avoided, and for several other useful comments that improved the paper.

Appendix: Cholesky updating

The Cholesky factor of $\mathbf{H}_{-k,-k}$ can be obtained from the Cholesky factorization $\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{H}$, by starting from the factorization $\mathbf{R}^\mathsf{T}_{\bullet,-k}\mathbf{R}_{\bullet,-k} = \mathbf{H}_{-k,-k}$, and noting that if $\mathbf{Q}$ is any appropriately dimensioned orthogonal matrix then $\mathbf{R}^\mathsf{T}_{\bullet,-k}\mathbf{Q}^\mathsf{T}\mathbf{Q}\mathbf{R}_{\bullet,-k} = \mathbf{H}_{-k,-k}$. Hence choosing $\mathbf{Q}$ so that $\mathbf{Q}\mathbf{R}_{\bullet,-k}$ is upper triangular, and discarding its final row, we have the Cholesky factor of $\mathbf{H}_{-k,-k}$. An appropriate $\mathbf{Q}$ can be constructed using Givens rotations (see Golub and Van Loan, 2013, §5.1.8), resulting in the following algorithm, where $\mathbf{R}_{\bullet,-k}$ is referred to as $\mathbf{R}$ to simplify notation, and loops with a negative range are skipped:

On input $\mathbf{R}$ is the $p \times p-1$ result of dropping column k of an upper triangular matrix.

For $i = k, \ldots, p-1$
  Set $\alpha = (R_{i,i}^2 + R_{i+1,i}^2)^{1/2}$, $c = R_{i,i}/\alpha$ and $s = R_{i+1,i}/\alpha$
  Set $R_{i,i} = \alpha$ and $R_{i+1,i} = 0$.
  For $j = i+1, \ldots, p-1$
    $r = R_{i,j}$.
    $R_{i,j} = cr + sR_{i+1,j}$.
    $R_{i+1,j} = -sr + cR_{i+1,j}$.
Drop the final row of R.

On exit $\mathbf{R}$ is the required $p \times p$ upper triangular Cholesky factor. This simple statement of the algorithm is efficient for row major storage, but for column major storage, as used in R, a column oriented version optimizes memory access, requiring storage of vectors of the c and s coefficients at each i for re-use with each column j.

References

Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*. Philadelphia: SIAM.

Golub, G. H. and C. F. Van Loan (2013). *Matrix computations* (4th ed.). Baltimore: Johns Hopkins University Press.

Harville, D. A. (1997). *Matrix Algebra from a Statistician's Perspective*. New York: Springer.

Martins, T. G., D. Simpson, F. Lindgren, and H. Rue (2013). Bayesian computing with INLA: new features. *Computational Statistics & Data Analysis 67*, 68–83.

Nocedal, J. and S. Wright (2006). *Numerical Optimization* (2nd ed.). New York: Springer Verlag.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003).*, pp. 20–22.

Plummer, M. (2014). *rjags: Bayesian Graphical Models using MCMC*. R package version 3-13.

Rue, H., S. Martino, and N. Chopin (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*(2), 319–392.

Rue, H., A. Riebler, S. H. Sørbye, J. B. Illian, D. P. Simpson, and F. K. Lindgren (2017). Bayesian computing with INLA: a review. *Annual Review of Statistics and Its Application 4*, 395–421.

Sanchez, S. M. and P. J. Sanchez (2005). Very large fractional factorial and central composite designs. *ACM Transactions on Modeling and Computer Simulation 15*(4), 362–377.

Shun, Z. and P. McCullagh (1995). Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society, Series B 57*(4), 749–760.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society, Series B 47*(1), 1–53.

Sørbye, S. H. and H. Rue (2011). Simultaneous credible bands for latent Gaussian models. *Scandinavian Journal of Statistics 38*(4), 712–725.

Umlauf, N., D. Adler, T. Kneib, S. Lang, and A. Zeileis (2015). Structured Additive Regression Models: An R Interface to BayesX. *Journal of Statistical Software 63(21)*, 1–46.

Wood, S. N. (2015). *Core Statistics*. Cambridge University Press.

Wood, S. N. (2017). *Generalized Additive Models: An Introduction with R* (2 ed.). Boca Raton, FL: CRC press.

Wood, S. N., N. Pya, and B. Säfken (2016). Smoothing parameter and model selection for general smooth models (with discussion). *Journal of the American Statistical Association 111*, 1548–1575.