Fast approximate inference with INLA: the past, the present and the future

Daniel Simpson, Finn Lindgren and Håvard Rue
Department of Mathematical Sciences
Norwegian University of Science and Technology
N-7491 Trondheim, Norway

May 15, 2011

Abstract

Latent Gaussian models are an extremely popular, flexible class of models. Bayesian inference for these models is, however, tricky and time consuming. Recently, Rue, Martino and Chopin introduced the Integrated Nested Laplace Approximation (INLA) method for deterministic fast approximate inference. In this talk we will outline the INLA approximation and its related R package. We will focus on using INLA for survival and point process models and demonstrate some of the new features. Finally we will discuss possible extensions for INLA.

1 Introduction

As the statistical understanding of applied scientists increases and new techniques deliver larger, more complicated data sets, applied statisticians are faced with increasingly complex models. Naturally, as the complexity of these models increase, it becomes harder and harder to perform inference. Appropriately, a great deal of effort has been expended on constructing numerical methods for performing approximate Bayesian inference. Undoubtably, the most popular family of approximate inference methods in Bayesian statistics is the class of Markov Chain Monte Carlo (MCMC) methods. These methods, which exploded into popularity in the mid 1980s and have remained at the forefront of Bayesian statistics ever since, with the basic framework being extended to cope with increasingly more complex problems.

The key advantage of MCMC methods is that, in their most vanilla incarnation, they are extremely simple to program. This simplicity, together with their incredible flexibility, has lead to the proliferation of these methods. Of course, there are problems: a single site auxiliary Gibbs sampler for spatial logistic regression is known to fail spectacularly. This is just the tip of the iceberg—for even mildly complicated models, it can be extremely difficult to construct a MCMC scheme that converges in a reasonable amount of time.

For large models, and especially spatial models, fast convergence isn't enough. Even if you could sample exactly from the posterior, sampling–based methods converge like $\mathcal{O}(N^{-1/2})$, where N is the number of samples, which suggests that you need 10^{2p} samples to get an error of around 10^{-p}. Clearly, if computing a single sample is even reasonably expensive, this cost will be prohibitive. In the best case, this means that MCMC schemes for large problems typically take hours or even days to deliver estimates that are only correct to three or four decimal places. Clearly this is less than ideal!

The only way around this efficiency problem is to consider alternatives to sampling-based methods. The first step in constructing an efficient approximate inference scheme is to greatly restrict the class of models that we will consider: it is naïve to expect that an efficient algorithm exists that will solve all of the problems that MCMC treats. With this in mind, we restrict our attention to the class of *latent Gaussian models*, which we define in three stages as

$$\begin{aligned} y_i|\mathbf{x} &\sim \pi(y_i|x_i) && \text{(Observation equation)} \\ \mathbf{x}|\boldsymbol{\theta} &\sim N(\boldsymbol{\mu}(\boldsymbol{\theta}), \mathbf{Q}(\boldsymbol{\theta})^{-1}) && \text{(Latent Gaussian field)} \\ \boldsymbol{\theta} &\sim \pi(\boldsymbol{\theta}) && \text{(Parameter model),} \end{aligned}$$

where $\mathbf{Q}(\boldsymbol{\theta})$ is the *precision matrix* (that is, the inverse of the covariance matrix) of the Gaussian random vector $\mathbf{x}$. In the interest of having a computable model, we will restrict $\mathbf{Q}$ to be either sparse or small enough that computing multiple factorisations is not an issue. These models cover a large chunk of classical statistical models, including dynamic linear models, stochastic volatility models, generalised linear (mixed) models, generalised additive (mixed) models, spline smoothing models, disease mapping, log-Gaussian Cox processes, model-based geostatistics, spatio-temporal models and survival analysis.

The Integrated Nested Laplace Approximation (INLA), builds upon the use of Laplace approximations, which were originally for approximating posterior distributions by Tierney and Kadane (1986). The first step in the INLA approximation is to perform a Laplace approximation to the joint posterior

$$\begin{aligned} \pi(\boldsymbol{\theta}|\mathbf{y}) &= \frac{\pi(\boldsymbol{\theta})\pi(\boldsymbol{x},\boldsymbol{\theta})\pi(\boldsymbol{y}|\boldsymbol{x})}{\pi(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})} \\ &\propto \frac{\pi(\boldsymbol{\theta})\pi(\boldsymbol{x},\boldsymbol{\theta})\pi(\boldsymbol{y}|\boldsymbol{x})}{\pi_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}, \end{aligned} \tag{1}$$

where $\pi_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ is the Gaussian approximation to $\pi(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ that matches the true distribution at the mode (Rue et al., 2009). The approximate posterior marginals for the non-Gaussian parameters can then be constructed through numerical integration as long as the dimension of $\boldsymbol{\theta}$ is not too large. The posterior marginals for the latent field $\pi(x_i|\boldsymbol{y})$ are constructed by computing a Laplace approximation to $\pi(x_i|\boldsymbol{\theta},\boldsymbol{y})$ and then integrating out against the approximate joint posterior for $\boldsymbol{\theta}|\boldsymbol{y}$. Full details of the approximation scheme can be found in Rue et al. (2009).

2 The `r-INLA` program

The INLA method was designed to be provide fast inference for a large class of practical Bayesian problems. In order to fulfil this aim, the `r-INLA` package was created as an `R` interface to the INLA program, which is itself written in `C`. The syntax for the `r-INLA` package is based on the inbuilt `glm` function in `R`, which highlights the effectiveness of the INLA method as a general solver for generalised linear (mixed) models. The `r-INLA` package is available from `http://r-inla.org`.

They key to the computational efficiency of the `r-INLA` program is that it is based on `GMRFLib`, a `C` library written by Håvard Rue for performing efficient computations on Gaussian Markov random fields. As such, `r-INLA` is particularly effective when the latent Gaussian field has the Markov property. This covers the case of spline smoothing (in any dimension), as well as conditional autoregressive models and some Matérn random fields (Lindgren et al., 2011). Such a latent field is specified through the `formula` mechanism in `R`.

To demonstrate the `r-INLA` package, let us consider some survival data for myeloid leukaemia cases in the north-west of England. The model is a Cox proportional hazard model, where the hazard depends linearly on the age and sex of the patient, smoothly on the white blood count (`wbc`) and an econometric covariate (`tpi`). Furthermore, it is assumed that there is a spatially correlated random effect, which takes into account which district the patient is in. The following code performs full Bayesian inference on the appropriate generalised additive mixed model in around seven seconds. The posterior mean spatial effect is shown in Figure 1

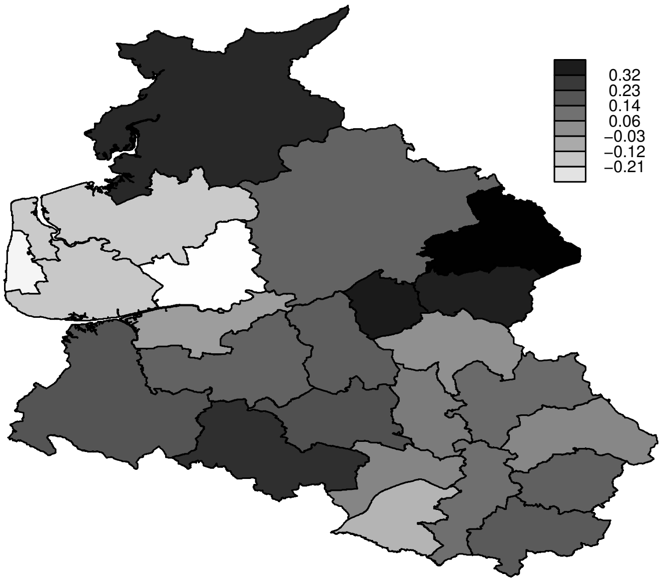

Figure 1: The posterior mean for the effect of district.

```
> data(Leuk)
> g = system.file("demodata/Leuk.graph", package = "INLA")
> formula = inla.surv(Leuk$time, Leuk$cens) ~ 1 + age + sex + f(inla.group(wbc),
+     model = "rw1") + f(inla.group(tpi), model = "rw2") + f(district,
+     model = "besag", graph.file = g)
> result = inla(formula, family = "coxph", data = Leuk)
```

3 New features

Since the original INLA paper, there have been a number of new developments. In this section, we outline some of the most recent additions to the `r-INLA` package.

Manipulating the likelihood The original INLA method was limited to observation models where each observation depended on one element of the latent Gaussian field. While this is commonly the case, this assumption is violated, for example, when the observed data consists of area averages of the latent field. In this case,

$$y_i|\boldsymbol{x} \sim \pi\left(y_i\Big|\sum_j a_{ij}x_j\right).$$

We further assume that the dependence of the data on the latent field is "local" in the sense that most elements of the "$\boldsymbol{A}$ matrix" are zero. With this assumption, everything stays Markovian and fast inference is still possible. This is implemented in the `r-INLA` program by modifying the `control.compute` parameter in the `r-INLA` function call:

```
> res = inla(formula, family = "...", data = ..., control.compute = list(A = amat))
```

Beyond relaxing this restriction to the class of models considered by the r-INLA program, there are a number of other new methods for building new models. The f() function, which r-INLA uses to specify random effects, has two new options: replicate and copy. The first option can be used to simply deal with the case where the likelihood requires independent replicates of the model with the same hyperparameters. The copy option is useful in situations where the latent field uses the same random field multiple times, possibly with different scalings.

Finally, r-INLA has been extended to include models where the data comes from different sources. In this case, different subsets of the data will require different likelihood functions. This can be programmed in r-INLA by re-writing the data as a matrix where the number of columns are equal to the number of likelihoods. In the case where there are two likelihoods, each containing n data points, this is achieved through the command

```
> Y = matrix(NA, N, 2)
> Y[1:n, 1] = y[1:n]
> Y[1:n + n, 2] = y[(n + 1):(2 * n)]
```

The r-INLA command is then modified appropriately by setting family = c("model1", "model2").

Survival models A class of models that were not considered in the original INLA paper were Bayesian survival models. The trick is to see Bayesian survival models as just another set of Latent Gaussian models. In some situations, this is straightforward, while at other times it requires data augmentation tricks, which are implemented in the inla.surv() function, demonstrated in Section 2. These methods are outlined in (Akerkar et al., 2010; Martino et al., 2010), which also discuss ways to deal with different types of censoring.

Stochastic partial differential equations A new method for constructing computationally efficient Gaussian random fields by taking advantage of the spatial Markov property was presented in a recent read paper by Lindgren et al. (2011). The idea is to use the fact that these fields can be represented as the solution to stochastic partial differential equations (SPDEs) to construct computationally efficient approximations to them. Beyond building computationally efficient approximations to standard spatial models, this method also allows for the construction of *new* classes of random fields with physically interpretable non-stationary. These models have been implemented in r-INLA . The following chunk of code fits a Bayesian spline through some noisy data points.

It begins by constructing a mesh on the unit square with vertices at the observation locations (points)

```
> bnd = inla.mesh.segment(matrix(c(0, 0, 1, 0, 1, 1, 0, 1), ncol = 2,
+     byrow = TRUE))
> mesh = inla.mesh.create(points, boundary = bnd, refine = list(max.edge = 0.1))
```

The second step is to construct the SPDE model

```
> spde = inla.spde.create(mesh, model = "imatern")
```

where imatern is the intrinsic matern model with $\nu = 1$, i.e. the spline smoothing model. Finally the formula is constructed and the inference is performed in the standard way:

```
> formula = y ~ f(data_points, model = spde) - 1
> r = inla(formula, family = "gaussian", data = list(y = y, data_points = mesh$idx$loc))
```

4 What the future holds

There are an almost endless number of ways that the INLA method r-INLA program can be extended. In this section we describe some of the new features that we are currently working on.

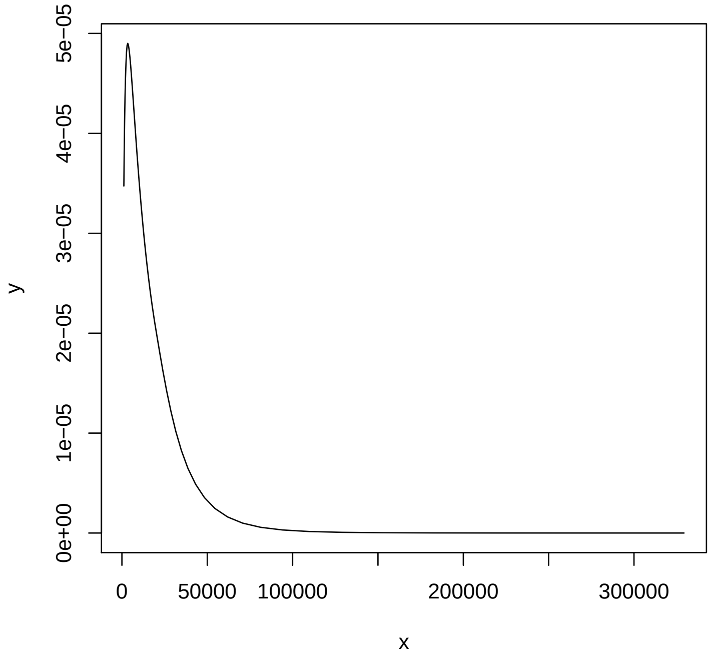

Figure 2: The precision for the latent Gaussian field is badly overestimated—the true value is $\phi = 1$.

Fixing "failures": global Gaussian approximations The Laplace approximation proceeds by fitting a Gaussian approximation around the mode of $\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$, however there are situations in which this is not the most appropriate approximation. For instance, if the true distribution is bimodal, a better 'fit' would be obtained by constructing a Gaussian approximation that *globally* approximates the distribution.

Another situation where these more global approximation would be of use is the following case of "failure". Consider the problem of approximating the latent random field for the following logistic regression model.

```
> n = 100
> eta = 1 + rnorm(n)
> p = exp(eta)/(1 + exp(eta))
> y = rbinom(n, size = 1, prob = p)
> bad.result = inla(y ~ 1 + f(num, model = "iid"), family = "binomial",
+     Ntrials = rep(1, n), data = list(y = y, num = c(1:100)))
```

Figure 2 shows the posterior for the precision of the random effect. INLA has clearly missed the correct precision, which was 1.

So what went wrong? Quite simply there is very little information in the data and hence the model is very prior sensitive. This sensitivity, combined with the vague prior that `r-INLA` uses as a default produced the nonsense results in Figure 2.

Kronecker product models In a number of applications, the precision matrix in the Gaussian random field can be written as a Kronecker product of two standard covariance matrices. A simple example of this is the separable space-time model constructed by using spatially correlated innovations in an AR(1) model:

$$\boldsymbol{x}_{t+1} = \phi \boldsymbol{x}_t + \boldsymbol{\epsilon}_t,$$

where ϕ is a scalar and $\boldsymbol{\epsilon} \sim N(0, \boldsymbol{Q}_\epsilon{}^{-1})$. In this case, the precision matrix is $\boldsymbol{Q} = \boldsymbol{Q}_{\mathrm{AR}(1)} \otimes \boldsymbol{Q}_\epsilon$, where $\otimes$ is the Kronecker product.

Due to the prevalence of Kronecker product models, it is desirable to add a Kronecker product mechanism to `r-INLA` . The general Kronecker product mechanism is currently in progress, but a number of special cases are already available in the code through the undocumented `group` feature. For example, a separable spatiotemporal SPDE model can be constructed using the command

```
> frm = y ~ f(loc, model = spde, group = time, control.group = list(model = "ar1"))
```

in which every observation `y` is assigned a location `loc` and a time `time`. At each time, the spatial points are linked by an SPDE model, while across the time periods, they evolve according to an AR(1) process.

Extending the SPDE methodology The grouping mechanism described above can be used to produce separable space-time models, that is models in which the covariance function can be factored into a purely spatial and a purely temporal component. In some situations, this type of separability is an unrealistic assumption and a great deal of research has gone into constructing classes of non-separable spatiotemporal covariance functions. An interesting property of SPDE models is that *any* model built with a sensible space-time partial differential operator will lead to a non-separable model. Furthermore, these models will inherit the good physical properties of the deterministic PDE models, such as causality and non-reversibility. This guarantees that the non-separability is *useful*, rather than simply present.

We are currently working to include the stochastic heat equation model

$$\frac{\partial}{\partial t}(\tau(s,t)x(s,t)) - \nabla \cdot (\boldsymbol{D}(s,t)\nabla(\tau(s,t)x(s,t))) + \nabla \cdot (\boldsymbol{b}(s,t)x(s,t)) + \kappa^2(s,t)x(s,t) = W(s,t),$$

where the noise process $W(s,t)$ is white in time, but correlated and Markovian in space. The challenge here is not simply placing the model into the `r-INLA` framework. This model includes temporally varying anisotropy and temporally varying drift, and therefore, even parameterising this model is an open problem.

Gamma frailty models: relaxing the Gaussian assumptions The assumption of Gaussian random effects is at the very heart of the INLA approximation. However, there are a number of situations in which this is not a realistic assumption. An example of this comes when incorporating frailty into Cox proportional hazard models. In these models, the hazard function for individual i is modelled as

$$h(t_i) = h_0 \nu_i \exp(\eta_i),$$

where η_i is a linear model containing covariates and ν_i is the frailty term, which models unobserved heterogeneity in the population. Clearly, if we take ν_i to be log-normal, the resulting model fits firmly in the standard INLA framework. Unfortunately, log-normal frailties are an uncommon model, typically the frailty term is taken to be gamma distributed. The question is, therefore, can we incorporate gamma frailty models into the INLA framework.

The solution to this problem comes in the guise of "importance sampling"–type decomposition:

$$\text{Gamma} = \underbrace{\text{LogNormal}}_{\text{“Prior”}} \times \underbrace{\frac{\text{Gamma}}{\text{LogNormal}}}_{\text{“Correction”}}.$$

With this type of formulation, it is possible to include gamma frailty models into the INLA framework.

This approach is not entirely satisfactory—although we can theoretically do this for any model suitably close to the log-normal (such as the log-t distribution), it is not particularly flexible. The aim of this work is to incorporate ideas from Bayesian nonparametrics to construct a class of suitable non-Gaussian random effects models that can be incorporated into this framework. This will massively increase the class of models for which INLA is available.

5 Conclusion

This article was finished on 15th May, 2011 and all of the information about INLA is correct at this time. This statement is necessary—INLA is still a project in active development. By the time you read this, some of the 'present' features will have moved into the 'past', and the 'future' features will be edging ever closer to inclusion. In fact, those who are interested can follow the progress of the INLA project at `http://inla.googlecode.com`, or by frequently updating the 'testing' version of INLA using the command

```
> inla.update(testing=TRUE)
```

This 'testing' version of INLA updates frequently and includes experimental interfaces to the newest features. This build also has the pleasant feature of matching with the documentation on `http://r-inla.org`!

The `r-INLA` project was created to provide an easy to use tool for performing Bayesian inference on latent Gaussian models. As such, the set of problems that `r-INLA` can solve is limited to those that someone has wanted to solve. There are any number of possible extensions not listed in the 'future' section that we are not currently considering because no one has asked for them yet. The lesson here is *if you want `r-INLA` to have a particular feature, observation model or prior model, you need to ask us!* The development of the INLA project is driven entirely by the research interests of the development team and the requests that we receive from the user community.

Acknowledgements

The INLA team is Håvard Rue, Sara Martino, Finn Lindgren, Daniel Simpson, Thiago Guerrera Martins and Rupali Akerkar.

References

R. Akerkar, S. Martino, and H. Rue. Implementing approximate Bayesian inference for survival analysis using integrated nested Laplace approximations. Technical report 1, Department of mathematical sciences, Norwegian University of Science and Technology, 2010.

F. Lindgren, J. Lindström, and H. Rue. An explicit link between Gaussian fields and Gaussian Markov random fields: The SPDE approach. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 2011.

S. Martino, R. Akerkar, and H. Rue. Approximate Bayesian inference for survival models. Technical report 3, Department of mathematical sciences, Norwegian University of Science and Technology, 2010.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B*, 71(2):319–392, 2009.

L. Tierney and J. B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.