

Leave Pima Indians Alone: Binary Regression as a Benchmark for Bayesian Computation

Nicolas Chopin and James Ridgway

Abstract. Whenever a new approach to perform Bayesian computation is introduced, a common practice is to showcase this approach on a binary regression model and datasets of moderate size. This paper discusses to which extent this practice is sound. It also reviews the current state of the art of Bayesian computation, using binary regression as a running example. Both sampling-based algorithms (importance sampling, MCMC and SMC) and fast approximations (Laplace, VB and EP) are covered. Extensive numerical results are provided, and are used to make recommendations to both end users and Bayesian computation experts. Implications for other problems (variable selection) and other models are also discussed.

Key words and phrases: Bayesian computation, expectation propagation, Markov chain Monte Carlo, sequential Monte Carlo, variational inference.

1. INTRODUCTION

The field of Bayesian computation seems hard to track these days as it is blossoming in many directions. MCMC (Markov chain Monte Carlo) remains the main approach, but it is no longer restricted to Gibbs sampling and Hastings–Metropolis, as it includes more advanced, physics-inspired methods, such as HMC (Hybrid Monte Carlo, Neal, 2011) and its variants (Girolami and Calderhead, 2011, Shahbaba et al., 2014, Hoffman and Gelman, 2014). On the other hand, there is also a growing interest for alternatives to MCMC, such as SMC (Sequential Monte Carlo, e.g., Del Moral, Doucet and Jasra, 2006), nested sampling (Skilling, 2006), or the fast approximations that originated from machine learning, such as VB (Variational Bayes; see, e.g., Bishop, 2006, Chapter 10), and EP (Expectation Propagation, Minka, 2001). Even Laplace approximation has resurfaced in particular thanks to

the INLA methodology (Rue, Martino and Chopin, 2009).

One thing, however, that all these approaches have in common is they are almost always illustrated by a binary regression example; see, for example, the aforementioned papers. In other words, binary regression models, such as probit or logit, are a de facto benchmark for Bayesian computation.

This remark leads to several questions. Are binary regression models a reasonable benchmark for Bayesian computation? Should they be used then to develop a “benchmark culture” in Bayesian computation, like in, for example, optimisation? And practically, which of these methods actually “works best” for approximating the posterior distribution of a binary regression model?

The objective of this paper is to answer these questions. As the title suggests, an important factor in our discussion will be the size of the considered dataset. In fact, one of our final recommendations will be to consider using datasets that are bigger than those often found in the literature (such as the popular Pima Indians dataset), but we shall return to this point in the conclusion.

We would also like to discuss *how* Bayesian computation algorithms should be compared. One obvious

Nicolas Chopin is Professor of Statistics at the ENSAE, Paris, 3 Avenue Pierre Larousse, 92245 Malakoff, France (e-mail: nicolas.chopin@ensae.fr). James Ridgway is a postdoctoral researcher at INRIA-Lille–Nord Europe, équipe SequeL, Parc Scientifique de la Haute-Borne, 40 avenue Halley, 59650 Villeneuve d’Ascq, France (e-mail: james.lp.ridgway@gmail.com).

criterion is the error versus CPU time trade-off; this implies discussing which posterior quantities one may need to approximate. A related point is whether the considered method comes with a simple way to evaluate the numerical error. Other criteria of interest are: (a) how easy to implement is the considered method? (b) how generic is it? (does changing the prior or the link function require a complete rewrite of the source code?) (c) to which extent does it require manual tuning to obtain good performances? (d) is it amenable to parallelisation? Points (a) and (b) are rarely discussed in statistics, but relate to the important fact that, the simpler the program, the easier it is to maintain, and to make it bug-free. Regarding point (c), we warn beforehand that, as a matter of principle, we shall refuse to manually tune an algorithm on a per dataset basis. Rather, we will discuss, for each approach, some (hopefully reasonable) general recipe for how to choose the tuning parameters. This has two motivations. First, human time is far more valuable than computer time: [Cook \(2014\)](#) mentions that one hour of CPU time is today three orders of magnitude less expensive than one hour of pay for a programmer (or similarly a scientist). Second, any method requiring too much manual tuning through trial and error may be practically of no use beyond a small number of experts.

Finally, we also hope this paper may serve as an up to date review of the state of Bayesian computation. We believe this review to be timely for a number of reasons. First, as already mentioned, because Bayesian computation seems to develop currently in several different directions. Second, and this relates to criterion (d), the current interest in parallel computation ([Lee et al., 2010](#), [Suchard et al., 2010](#)) may require a reassessment of Bayesian computational methods: method A may perform better than method B on a single core architecture, while performing much worse on a parallel architecture. Finally, although the phrase “big data” seems to be a tired trope already, it is certainly true that datasets are getting bigger and bigger, which in return means that statistical methods needs to be evaluated on bigger and bigger datasets.

The paper is structured as follows. Section 2 covers certain useful preliminaries on binary regression models. Section 3 discusses fast approximations, that is, deterministic algorithms that compute an approximation of the posterior, at a lower cost than sampling-based methods. Section 4 discusses “exact,” sampling-based methods. Section 5 is the most important part of the paper, as it contains an extensive numerical comparison of all these methods. Section 6 discusses variable

selection. Section 7 discusses our findings, and their implications for both end users and Bayesian computation experts. Section 8 discusses to which extent our conclusions apply to other models.

2. PRELIMINARIES: BINARY REGRESSION MODELS

2.1 Likelihood, Prior

The likelihood of a binary regression model has the generic expression

$$(2.1) \quad p(\mathcal{D}|\boldsymbol{\beta}) = \prod_{i=1}^{n_{\mathcal{D}}} F(y_i \boldsymbol{\beta}^T \mathbf{x}_i),$$

where the data \mathcal{D} consist of $n_{\mathcal{D}}$ responses $y_i \in \{-1, 1\}$ and $n_{\mathcal{D}}$ vectors \mathbf{x}_i of p covariates, and F is some CDF (cumulative distribution function) that transforms the linear form $y_i \boldsymbol{\beta}^T \mathbf{x}_i$ into a probability. Taking $F = \Phi$, the standard normal CDF, gives the probit model, while taking $F = L$, the logistic CDF, $L(x) = 1/(1 + e^{-x})$, leads to the logistic model. Other choices could be considered, such as, for example, the CDF of a Student distribution (robit model) to better accommodate outliers.

We follow [Gelman et al.’s \(2008\)](#) recommendation to standardise the predictors in a preliminary step: non-binary predictors have mean 0 and standard deviation 0.5, binary predictors have mean 0 and range 1 and the intercept (if present) is set to 1. This standardisation facilitates prior specification: one then may set up a “weakly informative” prior for $\boldsymbol{\beta}$, that is a proper prior that assigns a low probability that the marginal effect of one predictor is outside a reasonable range. Specifically, we shall consider two priors $p(\boldsymbol{\beta})$ in this work: (a) the default prior recommended by [Gelman et al. \(2008\)](#), a product of independent Cauchys with centre 0 and scale 10 for the constant predictor, 2.5 for all the other predictors (henceforth, the Cauchy prior); and (b) a product of independent Gaussians with mean 0 and standard deviation equal to twice the scale of the Cauchy prior (henceforth the Gaussian prior).

Of course, other priors could be considered, such as, for example, Jeffreys’ prior ([Firth, 1993](#)), or a Laplace prior ([Kabán, 2007](#)). Our main motivation for considering the two priors above is to determine to which extent certain Bayesian computation methods may be prior-dependent, either in their implementation (e.g., Gibbs sampling) or in their performance, or both. In particular, one may expect the Cauchy prior to be more difficult to deal with, given its heavy tails.

2.2 Data-Augmentation Formulation

An alternative way to define a binary regression model is by introducing latent variables as follows:

$$(2.2) \quad z_i = \boldsymbol{\beta}^T \mathbf{x}_i + n_i,$$

$$(2.3) \quad y_i = \text{sgn}(z_i),$$

where $\mathbf{z} = (z_1, \dots, z_{n_{\mathcal{D}}})^T$ is a vector of latent (unobserved) variables. One obtains probit (resp., logit) regression by taking $\epsilon_i \sim N(0, 1)$ [resp., $\epsilon_i \sim \text{Logistic}(0, 1)$].

Under this ‘‘data-augmentation’’ formulation, one recovers $p(\boldsymbol{\beta}|\mathcal{D})$ as the marginal distribution of the joint $p(\boldsymbol{\beta}, \mathbf{z}|\mathcal{D})$. Some of the approaches reviewed in this paper are based on this property: that is, they either sample from, or approximate in some way, $p(\boldsymbol{\beta}, \mathbf{z}|\mathcal{D})$.

2.3 Posterior Maximisation (Gaussian Prior)

We explain in this section how to quickly compute the mode, and the Hessian at the mode, of the posterior:

$$p(\boldsymbol{\beta}|\mathcal{D}) = \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{p(\mathcal{D})},$$

$$p(\mathcal{D}) = \int_{\mathbb{R}^d} p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta}) d\boldsymbol{\beta},$$

where $p(\boldsymbol{\beta})$ is one of the two priors presented in the previous section, and $p(\mathcal{D})$ is the marginal likelihood of the data (also known as the evidence). These quantities will prove useful later, in particular to tune certain of the considered methods.

The two first derivatives of the log-posterior density may be computed as

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}|\mathcal{D}) = \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}) + \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\mathcal{D}|\boldsymbol{\beta}),$$

$$\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \log p(\boldsymbol{\beta}|\mathcal{D}) = \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \log p(\boldsymbol{\beta})$$

$$+ \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \log p(\mathcal{D}|\boldsymbol{\beta}),$$

where

$$\frac{\partial}{\partial \boldsymbol{\beta}} \log p(\mathcal{D}|\boldsymbol{\beta}) = \sum_{i=1}^{n_{\mathcal{D}}} (\log F)'(y_i \boldsymbol{\beta}^T \mathbf{x}_i) y_i \mathbf{x}_i,$$

$$\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \log p(\mathcal{D}|\boldsymbol{\beta}) = \sum_{i=1}^{n_{\mathcal{D}}} (\log F)''(y_i \boldsymbol{\beta}^T \mathbf{x}_i) \mathbf{x}_i \mathbf{x}_i^T$$

and $(\log F)'$ and $(\log F)''$ are the two first derivatives of $\log F$. Provided that $\log F$ is concave, which is the case for probit and logit regressions, the Hessian of

the log-likelihood is clearly a negative definite matrix. Moreover, if we consider the Gaussian prior, then the Hessian of the log-posterior is also negative (as the sum of two negative matrices, as Gaussian densities are log-concave). We stick to the Gaussian prior for now.

This suggests the following standard approach to compute the MAP (maximum a posteriori) estimator, that is the point $\boldsymbol{\beta}_{\text{MAP}}$ that maximises the posterior density $p(\boldsymbol{\beta}|\mathcal{D})$: to use Newton–Raphson, that is, to iterate

$$(2.4) \quad \boldsymbol{\beta}_{(\text{new})} = \boldsymbol{\beta}_{(\text{old})} - \mathbf{H}^{-1} \left\{ \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}_{(\text{old})}|\mathcal{D}) \right\}$$

until convergence is reached; here, \mathbf{H} is Hessian of the log posterior at $\boldsymbol{\beta} = \boldsymbol{\beta}_{(\text{old})}$, as computed above. The iteration above corresponds to finding the zero of a local, quadratic approximation of the log-posterior. Newton–Raphson typically works very well (converges in a small number of iterations) when the function to maximise is concave.

We note two points in passing. First, one may obtain the MLE (maximum likelihood estimator) by simply taking $p(\boldsymbol{\beta}) = 1$ above (i.e., a Gaussian with infinite variance). But the MLE is not properly defined when complete separation occurs, that is, there exists a hyperplane that separates perfectly the two outcomes: $y_i \boldsymbol{\beta}_{\text{CS}}^T \mathbf{x}_i \geq 0$ for some $\boldsymbol{\beta}_{\text{CS}}$ and all $i \in 1 : N$. This remark gives an extra incentive for performing Bayesian inference, or at least MAP estimation, in cases where complete separation may occur, in particular when the number of covariates is large (Firth, 1993, Gelman et al., 2008).

Variants of Newton–Raphson may be obtained by adapting automatically the step size [e.g., update is $\boldsymbol{\beta}_{(\text{new})} = \boldsymbol{\beta}_{(\text{old})} - \lambda \mathbf{H}^{-1} \{ \frac{\partial}{\partial \boldsymbol{\beta}} \log p(\boldsymbol{\beta}_{(\text{old})}|\mathcal{D}) \}$, and step size λ is determined by line search] or replacing the Hessian \mathbf{H} by some approximation. Some of these algorithms such as IRLS (iterated reweighted least squares) have a nice statistical interpretation. For our purposes, however, these variants seem to show roughly similar performance, so we will stick to the standard version of Newton–Raphson.

2.4 Posterior Maximisation (Cauchy Prior)

The log-density of the Cauchy prior is not concave:

$$\log p(\boldsymbol{\beta}) = - \sum_{j=1}^p \log(\pi \sigma_j) - \sum_{j=1}^p \log(1 + \beta_j^2 / \sigma_j^2)$$

for scales σ_j chosen as explained in Section 2.1. Hence, the corresponding log-posterior is no longer

guaranteed to be concave, which in turn means that Newton–Raphson might fail to converge.

However, we shall observe that, for most of the datasets considered in this paper, Newton–Raphson does converge quickly even for our Cauchy prior. In each case, we used as starting point for the Newton–Raphson iterations the OLS (ordinary least square) estimate. We suspect what happens is that, for most standard datasets, the posterior derived from a Cauchy prior remains log-concave, at least in a region that encloses the MAP estimator and our starting point.

2.5 Connection with PAC-Bayesian Classification

The focus on this paper is on Bayesian inference for well defined statistical models, such as probit or logit. Consider, however, for the sake of the argument, replacing likelihood (2.1) by pseudo-likelihood:

$$(2.5) \quad p(\mathcal{D}|\boldsymbol{\beta}) = \exp\{-\lambda r(\boldsymbol{\beta})\},$$

where $\lambda > 0$, and $r(\boldsymbol{\beta})$ is a certain empirical risk function, such as the misclassification rate:

$$r(\boldsymbol{\beta}) = \frac{1}{n_{\mathcal{D}}} \sum_{i=1}^{n_{\mathcal{D}}} \mathbb{1}\{y_i \mathbf{x}_i^T \boldsymbol{\beta} < 0\}.$$

In addition, consider the pseudo-posterior $p(\boldsymbol{\beta}|\mathcal{D}) \propto p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})$. By taking the expectation $\boldsymbol{\beta}_{\text{post}}$ (or some similar quantity) of this pseudo-posterior, one obtains a linear classifier (i.e., function $\mathbf{x} \rightarrow \text{sgn}\{\mathbf{x}^T \boldsymbol{\beta}_{\text{post}}\}$) that should achieve a small misclassification rate.

What we have outlined above is the PAC (Probably Approximately Correct)-Bayesian approach to classification, which originates from [Shawe-Taylor and Williamson \(1997\)](#), [McAllester \(1998\)](#), [Catoni \(2004\)](#); see also [Catoni \(2007\)](#) for a general introduction, and [Bissiri, Holmes and Walker \(2013\)](#) for an interesting Bayesian perspective on this type of approach. Of course, a different risk function could be considered in (2.5), such as, for example, the AUC (area under curve) criterion ([Ridgway et al., 2014](#)).

The discussion of this paper will also apply to some extent to PAC-Bayesian classification. Note, however that, for risk functions such as the misclassification rate, the posterior log-density is neither concave nor differentiable. Hence posterior maximisation is no longer an option. We will return to this point later on.

3. FAST APPROXIMATION METHODS

This section discusses fast approximation methods, that is, methods that are deterministic, fast (compared

to sampling-based methods), but which come with approximation errors which are difficult to assess. These methods include the Laplace approximation, which was popular in statistics before the advent of MCMC methods, but also more recent Machine Learning methods, such as EP (Expectation Propagation, [Minka, 2001](#)), and VB (Variational Bayes, e.g., [Bishop, 2006](#), Chapter 10).

Concretely, we will focus on the approximation of the following posterior quantities: the marginal likelihood $p(\mathcal{D})$, as this may be used in model choice; and the marginal distributions $p(\beta_i|\mathcal{D})$ for each component β_i of $\boldsymbol{\beta}$. Clearly, these are the most commonly used summaries of the posterior distribution, and other quantities, such as the posterior expectation of $\boldsymbol{\beta}$, may be directly deduced from them.

Finally, one should bear in mind that such fast approximations may be used as a preliminary step to calibrate an exact, more expensive method, such as those described in Section 4.

3.1 Laplace Approximation

The Laplace approximation is based on a Taylor expansion of the posterior log-density around the mode $\boldsymbol{\beta}_{\text{MAP}}$:

$$\begin{aligned} \log p(\boldsymbol{\beta}|\mathcal{D}) &\approx \log p(\boldsymbol{\beta}_{\text{MAP}}|\mathcal{D}) \\ &\quad - \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}})^T \mathbf{Q}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}}), \end{aligned}$$

where $\mathbf{Q} = -\mathbf{H}$, that is, minus the Hessian of $\log p(\boldsymbol{\beta}|\mathcal{D})$ at $\boldsymbol{\beta} = \boldsymbol{\beta}_{\text{MAP}}$; recall that we explained how to compute these quantities in Section 2.3. One may deduce a Gaussian approximation of the posterior by simply exponentiating the equation above, and normalising:

$$(3.1) \quad \begin{aligned} q_L(\boldsymbol{\beta}) &= N_p(\boldsymbol{\beta}; \boldsymbol{\beta}_{\text{MAP}}, \mathbf{Q}^{-1}) \\ &:= (2\pi)^{-p/2} |\mathbf{Q}|^{1/2} \\ &\quad \cdot \exp\left\{-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}})^T \mathbf{Q}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{MAP}})\right\}. \end{aligned}$$

In addition, since for any $\boldsymbol{\beta}$,

$$p(\mathcal{D}) = \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\mathcal{D})}$$

one obtains an approximation to the marginal likelihood $p(\mathcal{D})$ as follows:

$$p(\mathcal{D}) \approx Z_L(\mathcal{D}) := \frac{p(\boldsymbol{\beta}_{\text{MAP}})p(\mathcal{D}|\boldsymbol{\beta}_{\text{MAP}})}{(2\pi)^{-p/2} |\mathbf{Q}|^{1/2}}.$$

From now on, we will refer to this particular Gaussian approximation q_L as the Laplace approximation, even

if this phrase is sometimes used in statistics for higher-order approximations, as discussed in the next section. We defer to Section 3.6 the discussion of the advantages and drawbacks of this approximation scheme.

3.2 Improved Laplace, Connection with INLA

Consider the marginal distributions $p(\beta_j|\mathcal{D}) = \int p(\boldsymbol{\beta}|\mathcal{D}) d\boldsymbol{\beta}_{-j}$ for each component β_j of $\boldsymbol{\beta}$, where $\boldsymbol{\beta}_{-j}$ is $\boldsymbol{\beta}$ minus β_j . A first approximation may be obtained by simply computing the marginals of the Laplace approximation q_L . An improved (but more expensive) approximation may be obtained from

$$p(\beta_j|\mathcal{D}) \propto \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{p(\boldsymbol{\beta}_{-j}|\beta_j, \mathcal{D})},$$

which suggests to choose a fine grid of β_j values [deduced for instance from $q_L(\boldsymbol{\beta})$], and for each β_j value, compute a Laplace approximation of $p(\boldsymbol{\beta}_{-j}|\beta_j, \mathcal{D})$, by computing the mode $\hat{\boldsymbol{\beta}}_{-j}(\beta_j)$ and the Hessian $\hat{H}(\boldsymbol{\beta}_j)$ of $\log p(\boldsymbol{\beta}_{-j}|\beta_j, \mathcal{D})$, and then approximate (up to a constant)

$$p(\beta_j|\mathcal{D}) \approx q_{\text{IL}}(\beta_j) \propto \frac{p(\hat{\boldsymbol{\beta}}(\beta_j))p(\mathcal{D}|\hat{\boldsymbol{\beta}}(\beta_j))}{|\hat{H}(\boldsymbol{\beta}_j)|^{1/2}},$$

where $\hat{\boldsymbol{\beta}}(\beta_j)$ is the vector obtained by inserting β_j at position i in $\hat{\boldsymbol{\beta}}_{-j}(\beta_j)$, and IL stands for ‘‘Improved Laplace.’’ One may also deduce posterior expectations of functions of β_j in this way. See also Tierney and Kadane (1986), Tierney, Kass and Kadane (1989) for higher order approximations for posterior expectations.

We note in passing the connection to the INLA scheme of Rue, Martino and Chopin (2009). INLA applies to posteriors $p(\boldsymbol{\theta}, \mathbf{x}|\mathcal{D})$ where \mathbf{x} is a latent variable such that $p(\mathbf{x}|\boldsymbol{\theta}, \mathcal{D})$ is close to a Gaussian, and $\boldsymbol{\theta}$ is a low-dimensional hyper-parameter. It constructs a grid of $\boldsymbol{\theta}$ -values, and for each grid point $\boldsymbol{\theta}_j$, it computes an improved Laplace approximation of the marginals of $p(\mathbf{x}|\boldsymbol{\theta}_j, \mathcal{D})$. In our context, $\boldsymbol{\beta}$ may be identified to \mathbf{x} , $\boldsymbol{\theta}$ to an empty set, and INLA reduces to the improved Laplace approximation described above.

3.3 The EM Algorithm of Gelman et al. (2008) (Cauchy Prior)

Gelman et al. (2008) recommend against the Laplace approximation for a Student prior (of which our Cauchy prior is a special case), because, as explained in Section 2.4, the corresponding log-posterior is not guaranteed to be concave, and this might prevent Newton–Raphson to converge. In our simulations, however, we found the Laplace approximation to work

reasonably well for a Cauchy prior. We now briefly describe the alternative approximation scheme proposed by Gelman et al. (2008) for Student priors, which we call for convenience Laplace-EM.

Laplace-EM is based on the well-known representation of a Student distribution, $\beta_j|\sigma_j^2 \sim N_1(0, \sigma_j^2)$, $\sigma_j^2 \sim \text{Inv-Gamma}(v/2, s_j v/2)$; take $v = 1$ to recover our Cauchy prior. Conditional on $\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_p^2)$, the prior on $\boldsymbol{\beta}$ is Gaussian, hence, for a fixed $\boldsymbol{\sigma}^2$ one may implement Newton–Raphson to maximise the log-density of $p(\boldsymbol{\beta}|\boldsymbol{\sigma}^2, \mathcal{D})$, and deduce a Laplace (Gaussian) approximation of the same distribution.

Laplace-EM is an approximate EM (Expectation Maximisation, Dempster, Laird and Rubin, 1977) algorithm, which aims at maximising in $\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_p^2)$ the marginal posterior distribution $p(\boldsymbol{\sigma}^2|\mathcal{D}) = \int p(\boldsymbol{\sigma}^2, \boldsymbol{\beta}|\mathcal{D}) d\boldsymbol{\beta}$. Each iteration involves an expectation with respect to the intractable conditional distribution $p(\boldsymbol{\beta}|\boldsymbol{\sigma}^2, \mathcal{D})$, which is Laplace approximated, using a single Newton–Raphson iteration. When this approximate EM algorithm has converged to some value $\boldsymbol{\sigma}_*^2$, one more Newton–Raphson iteration is performed to compute a final Laplace approximation of $p(\boldsymbol{\beta}|\boldsymbol{\sigma}_*^2, \mathcal{D})$, which is then reported as a Gaussian approximation to the posterior. We refer the readers to Gelman et al. (2008) for more details on Laplace-EM.

3.4 Variational Bayes

The general principle behind VB (Variational Bayes) is to approximate a posterior π by the member of a given family \mathcal{F} which is closest to π in Kullback–Leibler divergence:

$$(3.2) \quad q_{\text{VB}} = \arg \min_{q \in \mathcal{F}} \mathcal{K}(\pi \| q)$$

$$\text{with } \mathcal{K}(\pi \| q) = \int q \log\left(\frac{q}{\pi}\right).$$

The choice of \mathcal{F} is crucial: it must be large enough to make the approximation error small, while at the same time it must be such that the minimisation above is tractable.

For models involving latent variables \mathbf{z} , which is the case for binary regression models under the data-augmentation formulation (2.2)–(2.3), the standard ‘‘mean field’’ approach is to approximate the joint posterior $p(\boldsymbol{\beta}, \mathbf{z}|\mathcal{D})$, and to set \mathcal{F} to the set of distributions for $(\boldsymbol{\beta}, \mathbf{z})$ that factorises as follows: $q(\boldsymbol{\beta}, \mathbf{z}) = q_{\boldsymbol{\beta}}(\boldsymbol{\beta})q_{\mathbf{z}}(\mathbf{z})$. This makes it possible to minimise (3.2) using fixed-point iterations. Unfortunately, the solution is typically much more concentrated (i.e., has a smaller

variance) than the true posterior. [Consonni and Marin \(2008\)](#) give numerical evidence of this phenomenon for probit regression and a Gaussian prior. For more background on mean field VB, see Chapter 10 of [Bishop \(2006\)](#), and for recent developments see [Hoffman and Blei \(2014\)](#). Early references on mean VB include [Attias \(1999\)](#) and [Ghahramani and Beal \(2000\)](#).

Another VB approach is to approximate directly $p(\boldsymbol{\beta}|\mathcal{D})$, and to set \mathcal{F} to a parametric family of dimension p (e.g., the set of Gaussian distributions). Renewed interest in this parametric VB approach (e.g., [Oppen and Archambeau, 2009](#)) is due in part to the development of more powerful optimisation techniques to solve (3.2), such as stochastic gradient descent (SGD) ([Hoffman et al., 2013](#)) and optimal convex solvers ([Khan et al., 2013](#), [Alquier, Ridgway and Chopin, 2016](#)). Gradient descent although efficient is not always available in closed form for the VB objective. [Ranganath, Gerrish and Blei \(2014\)](#) and [Rezende, Mohamed and Wierstra \(2014\)](#) propose to write the gradient as an expectation and use a version of SGD. For logistic regression (with Gaussian prior) other tricks apply, see [Khan et al. \(2013\)](#).

We also mention in passing that [Alquier, Ridgway and Chopin \(2016\)](#) found that such a parametric VB approach works well for PAC-Bayesian classification, that is, for a posterior corresponding to pseudo-likelihood (2.5). However, since Gaussian VB does not seem generally applicable to binary regression models, we will not include it in our comparisons, and focus instead on Laplace and EP (see next section).

For recent reviews of variational inference, see [Wainwright and Jordan \(2008\)](#) and [Blei, Kucukelbir and McAuliffe \(2016\)](#). The former also gives an interesting unifying view of VB and EP (see next section). For extensions of VB to deal with larger datasets and more complex models see, for example, [Hoffman et al. \(2013\)](#), [Kingma and Welling \(2013\)](#), [Gregor et al. \(2015\)](#).

3.5 Expectation Propagation

Like Laplace and Gaussian VB, Expectation Propagation (EP, [Minka, 2001](#)) generates a Gaussian approximation of the posterior, but it is based on different ideas. The consensus in machine learning seems to be that EP provides a better approximation than Laplace (e.g., [Nickisch and Rasmussen, 2008](#)); the intuition being that Laplace is “too local” (i.e., it fitted so as to match closely the posterior around the mode), while EP is able to provide a global approximation to the posterior.

Starting from the decomposition of the posterior as product of $(n_{\mathcal{D}} + 1)$ factors:

$$p(\boldsymbol{\beta}|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \prod_{i=0}^{n_{\mathcal{D}}} l_i(\boldsymbol{\beta}),$$

$$l_i(\boldsymbol{\beta}) = F(y_i \boldsymbol{\beta}^T \mathbf{x}_i) \quad \text{for } i \geq 1,$$

and l_0 is the prior, $l_0(\boldsymbol{\beta}) = p(\boldsymbol{\beta})$, EP computes iteratively a parametric approximation of the posterior with the same structure

$$(3.3) \quad q_{\text{EP}}(\boldsymbol{\beta}) = \prod_{i=0}^{n_{\mathcal{D}}} \frac{1}{Z_i} q_i(\boldsymbol{\beta}).$$

Taking q_i to be an unnormalised Gaussian density written in natural exponential form

$$q_i(\boldsymbol{\beta}) = \exp \left\{ -\frac{1}{2} \boldsymbol{\beta}^T \mathbf{Q}_i \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{r}_i \right\},$$

one obtains for q_{EP} a Gaussian with natural parameters $\mathbf{Q} = \sum_{i=0}^n \mathbf{Q}_i$ and $\mathbf{r} = \sum_{i=0}^n \mathbf{r}_i$; note that the more standard parametrisation of Gaussians may be recovered by taking

$$\boldsymbol{\Sigma} = \mathbf{Q}^{-1}, \quad \boldsymbol{\mu} = \mathbf{Q}^{-1} \mathbf{r}.$$

Other exponential families could be considered for q and the q_i 's (see, e.g., [Seeger, 2005](#)), but Gaussian approximations seems the most natural choice here.

An EP iteration consists in updating one factor q_i , or equivalently $(Z_i, \mathbf{Q}_i, \mathbf{r}_i)$, while keeping the other factors as fixed, by moment matching between the hybrid distribution

$$h(\boldsymbol{\beta}) \propto l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta})$$

and the global approximation q defined in (3.3): compute

$$Z_h = \int l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta}) d\boldsymbol{\beta},$$

$$\boldsymbol{\mu}_h = \frac{1}{Z_h} \int \boldsymbol{\beta} l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta}) d\boldsymbol{\beta},$$

$$\boldsymbol{\Sigma}_h = \frac{1}{Z_h} \int \boldsymbol{\beta} \boldsymbol{\beta}^T l_i(\boldsymbol{\beta}) \prod_{j \neq i} q_j(\boldsymbol{\beta}) d\boldsymbol{\beta}$$

and set

$$\mathbf{Q}_i = \boldsymbol{\Sigma}_h^{-1} - \mathbf{Q}_{-i}, \quad \mathbf{r}_i = \boldsymbol{\Sigma}_h^{-1} \boldsymbol{\mu}_h - \mathbf{r}_{-i},$$

$$\log Z_i = \log Z_h - \Psi(\mathbf{r}, \mathbf{Q}) + \Psi(\mathbf{r}_{-i}, \mathbf{Q}_{-i}),$$

where $\mathbf{r}_{-i} = \sum_{j \neq i} \mathbf{r}_j$, $\mathbf{Q}_{-i} = \sum_{j \neq i} \mathbf{Q}_j$, and $\psi(\mathbf{r}, \mathbf{Q})$ is the normalising constant of a Gaussian distribution with natural parameters (\mathbf{r}, \mathbf{Q}) ,

$$\begin{aligned} \psi(\mathbf{r}, \mathbf{Q}) &= \int_{\mathbb{R}^p} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^T \mathbf{Q} \boldsymbol{\beta} + \boldsymbol{\beta}^T \mathbf{r}\right\} d\boldsymbol{\beta} \\ &= -\frac{1}{2} \log |\mathbf{Q}/2\pi| + \frac{1}{2} \mathbf{r}^T \mathbf{Q} \mathbf{r}. \end{aligned}$$

In practice, EP proceeds by looping over sites, updating each one in turn until convergence is achieved.

To apply EP to binary regression models, two points must be addressed. First, how to compute the hybrid moments? For the probit model, these moments may be computed exactly, see the supplemental article (Chopin and Ridgway, 2016), while for the other link function (such as logistic), numerical (one-dimensional) quadrature may be used.

Second, how to deal with the prior? If the prior is Gaussian, one may simply set q_0 to the prior, and never update q_0 in the course of the algorithm. For a Cauchy prior, q_0 is simply treated as an extra site.

Supporting theory for EP appeared only recently (Dehaene and Barthelmé, 2015a, 2015b); the former reference shows that, under certain conditions, the EP expectation is at distance $\mathcal{O}(n_{\mathcal{D}}^{-2})$ of the true posterior expectation [while Laplace is at distance $\mathcal{O}(n_{\mathcal{D}}^{-1})$]. This seems to explain why EP is typically more accurate than Laplace.

The main drawback of EP is that it often requires more expertise than Laplace to make it work. First, for a given model, the hybrid moments mentioned above may be difficult to compute. Second, in certain situations, site updates may generate a negative matrix \mathbf{Q}_i ; one may skip this site, but at the risk of an increased approximation error (if the site keeps on being skipped at subsequent iterations). Third, EP sometimes fail to converge; in particular when the target density is not log-concave (Seeger, 2005, Seeger, Gerwinn and Bethge, 2007). In that case, it may help to use fractional updates (Minka, 2004), where only a fraction $\alpha \in (0, 1)$ of site parameters $(Z_i, \mathbf{Q}_i, \mathbf{r}_i)$ is updated, but then choosing α requires some trial and error.

That said, in our experience binary regression models leads to well-behaved posteriors for which EP works well without much hassle.

3.6 Complexity of the Different Approximation Schemes

In our context, Laplace has complexity $\mathcal{O}(n_{\mathcal{D}} + p^3)$ per iteration, while EP has complexity $\mathcal{O}(n_{\mathcal{D}} p^2)$.

The p^3 term for Laplace comes from the fact that (2.4) involves solving a linear system. EP site updates involve a rank-one update of the precision matrix \mathbf{Q} , which may be performed in $\mathcal{O}(p^2)$ by using Woodbury’s formula. The overall complexity of both algorithms is harder to establish, as little is known on the number of iterations needed for EP to achieve a given error. Empirical evidence suggests however that EP is more expensive than Laplace on small enough datasets.

This remark may be mitigated as follows. First, one may modify EP so as to update the global approximation only at the end of each iteration (complete pass over the data). The resulting algorithm (van Gerven et al., 2010) may be easily implemented on parallel hardware: simply distribute the $n_{\mathcal{D}}$ factors over the processors. Even without parallelisation, parallel EP requires only one single matrix inversion per iteration.

Second, the “improved Laplace” approximation for the marginals described in Section 3.1 performs a large number of basic Laplace approximations, so its speed advantage compared to standard EP essentially vanishes.

Points that remain in favour of Laplace is that it is simpler to implement than EP, and the resulting code is very generic: adapting to either a different prior, or a different link function [choice of F in (2.1)], is simply a matter of writing a function that evaluates the corresponding function. We have seen that such an adaptation requires the user more work in EP, although to be fair the general structure of the algorithm is not model-dependent. On the other hand, we shall see that EP is often more accurate, and works in more examples, than Laplace; this is especially the case for the Cauchy prior.

More generally, one sees that the number of covariates p is more critical than the number of instances $n_{\mathcal{D}}$ in determining how “big” (how time-intensive to process) is a given dataset. This will be a recurring point in this paper.

4. EXACT METHODS

We now turn to sampling-based methods, which are “exact,” at least in the limit: one may make the approximation error as small as desired, by running the corresponding algorithm for long enough. (As mentioned by a referee, “long enough” may be practically impossible for certain complex models.) We will see that most of these algorithms require for good performance some form of calibration, which in turn requires some preliminary knowledge on the shape of the posterior distribution. Since the approximation methods covered

in the previous section are faster by orders of magnitude than sampling-based methods, we will assume that a Gaussian approximation $q(\boldsymbol{\beta})$ (say, obtained by Laplace or EP) has been computed in a preliminary step.

4.1 Our Gold Standard: Importance Sampling

Let $q(\boldsymbol{\beta})$ denote a generic approximation of the posterior $p(\boldsymbol{\beta}|\mathcal{D})$. Importance sampling (IS) is based on the trivial identity

$$\begin{aligned} p(\mathcal{D}) &= \int p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})d\boldsymbol{\beta} \\ &= \int q(\boldsymbol{\beta})\frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{q(\boldsymbol{\beta})}d\boldsymbol{\beta}, \end{aligned}$$

which leads to the following recipe: sample $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_N \sim q$, then compute as an estimator of $p(\mathcal{D})$

$$(4.1) \quad Z_N = \frac{1}{N} \sum_{n=1}^N w(\boldsymbol{\beta}_n), \quad w(\boldsymbol{\beta}) := \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{q(\boldsymbol{\beta})}.$$

In addition, since

$$\int \varphi(\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathcal{D})d\boldsymbol{\beta} = \frac{\int \varphi(\boldsymbol{\beta})q(\boldsymbol{\beta})w(\boldsymbol{\beta})d\boldsymbol{\beta}}{\int q(\boldsymbol{\beta})w(\boldsymbol{\beta})d\boldsymbol{\beta}}$$

one may approximate any posterior moment as

$$(4.2) \quad \varphi_N = \frac{\sum_{n=1}^N w(\boldsymbol{\beta}_n)\varphi(\boldsymbol{\beta}_n)}{\sum_{n=1}^N w(\boldsymbol{\beta}_n)}.$$

Approximating posterior marginals is also straightforward; one may for instance use kernel density estimation on the weighted sample $(\boldsymbol{\beta}_n, w(\boldsymbol{\beta}_n))_{n=1}^N$.

Concerning the choice of q , we will restrict ourselves to the Gaussian approximations generated either from Laplace or EP algorithm. It is sometimes recommended to use a Student distribution instead, as a way to ensure that the variance of the above estimators is finite, but we did not observe any benefit for doing so in our simulations.

It is of course a bit provocative to call IS our gold standard, as it is sometimes perceived as an obsolete method. We would like to stress however that IS is hard to beat relative to most of the criteria laid out in the [Introduction](#):

- Because it is based on IID sampling, assessing the Monte Carlo error of the above estimators is trivial: for example, the variance of Z_N may be estimated as N^{-1} times the empirical variance of the

weights $w(\boldsymbol{\beta}_n)$. The auto-normalised estimator (4.2) has asymptotic variance

$$\begin{aligned} \mathbb{E}_q[w(\boldsymbol{\beta})^2\{\varphi(\boldsymbol{\beta}) - \mu(\varphi)\}^2], \\ \mu(\varphi) = \int \varphi(\boldsymbol{\beta})p(\boldsymbol{\beta}|\mathcal{D})d\boldsymbol{\beta}, \end{aligned}$$

which is also trivial to approximate from the simulated $\boldsymbol{\beta}_n$'s.

- Other advantages brought by IID sampling are: (a) importance sampling is easy to parallelize; and (b) importance sampling is amenable to QMC (Quasi-Monte Carlo) integration, as explained in the following section.
- Importance sampling offers an approximation of the marginal likelihood $p(\mathcal{D})$ at no extra cost.
- Code is simple and generic.

Of course, what remains to determine is whether importance sampling does well relative to our main criterion, that is, error versus CPU trade-off. We do know that IS suffers from a curse of dimensionality: take both q and the target density π to be the density of IID distributions: $q(\boldsymbol{\beta}) = \prod_{j=1}^p q_1(\beta_j)$, $\pi(\boldsymbol{\beta}) = \prod_{j=1}^p \pi_1(\beta_j)$; then it is easy to see that the variance of the weights grows exponentially with p . Thus, we expect IS to collapse when p is too large; meaning that a large proportion of the $\boldsymbol{\beta}_n$ gets a negligible weight. On the other hand, for small to moderate dimensions, we will observe surprising good results; see Section 5. We will also present below a SMC algorithm that automatically reduces to IS when IS performs well, while doing something more elaborate in more difficult scenarios. Finally, we also note that [Marin and Robert \(2011\)](#) observed that importance sampling (with a proposal set to the Laplace approximation) outperforms several other methods for approximating the marginal likelihood of a probit model (and the Pima indians dataset).

The standard way to assess the weight degeneracy is to compute the effective sample size ([Kong, Liu and Wong, 1994](#)),

$$\text{ESS} = \frac{\{\sum_{n=1}^N w(\boldsymbol{\beta}_n)\}^2}{\sum_{n=1}^N w(\boldsymbol{\beta}_n)^2} \in [1, N],$$

which roughly approximates how many simulations from the target distribution would be required to produce the same level of error. In our simulations, we will compute instead the efficiency factor EF, which is simply the ratio $\text{EF} = \text{ESS} / N$.

4.2 Improving Importance Sampling by Quasi-Monte Carlo

Quasi-Monte Carlo may be seen as an elaborate variance reduction technique: starting from the Monte Carlo estimators Z_N and φ_N [see (4.1) and (4.2)], one may re-express the simulated vectors as functions of uniform variates \mathbf{u}_n in $[0, 1]^d$; for instance,

$$\boldsymbol{\beta}_n = \boldsymbol{\mu} + \mathbf{C}\boldsymbol{\zeta}_n, \quad \boldsymbol{\zeta}_n = \boldsymbol{\Phi}^{-1}(\mathbf{u}_n),$$

where $\boldsymbol{\Phi}^{-1}$ is Φ^{-1} , the $N(0, 1)$ inverse CDF, applied component-wise. Then one replaces the N vectors \mathbf{u}_n by a low-discrepancy sequence; that is, a sequence of N vectors that spread more evenly over $[0, 1]^d$; for example, a Halton or a Sobol' sequence. Under appropriate conditions, QMC error converges at rate $\mathcal{O}(N^{-1+\epsilon})$, for any $\epsilon > 0$, to be compared with the standard Monte Carlo rate $\mathcal{O}_p(N^{-1/2})$. We refer to Lemieux (2009) for more background on QMC, as well as how to construct QMC sequences.

Oddly enough, the possibility to use QMC in conjunction with importance sampling is very rarely mentioned in the literature; see, however, Hörmann and Leydold (2005). More generally, QMC seems often overlooked in statistics. We shall see, however, that this simple IS-QMC strategy often performs very well.

One drawback of IS-QMC is that we lose the ability to evaluate the approximation error in a simple manner. A partial remedy is to use randomised Quasi-Monte Carlo (RQMC), that is, the \mathbf{u}_n are generated in such a way that (a) with probability one, $\mathbf{u}_{1:N}$ is a QMC point set; and (b) each vector \mathbf{u}_n is marginally sampled from $[0, 1]^d$. Then QMC estimators that are empirical averages, such as $Z_N = N^{-1} \sum_{n=1}^N w(\boldsymbol{\beta}_n)$ become unbiased estimators, and their error may be assessed through the empirical variance over repeated runs. Technically, estimators that are ratios of QMC averages, such as φ_N , are not unbiased, but for all practical purposes their bias is small enough that assessing error through empirical variances over repeated runs remains a reasonable approach.

4.3 MCMC

The general principle of MCMC (Markov chain Monte Carlo) is to simulate a Markov chain that leaves invariant the posterior distribution $p(\boldsymbol{\beta}|\mathcal{D})$; see Robert and Casella (2004) for a general overview. Often mentioned drawbacks of MCMC simulation are (a) the difficulty to parallelize such algorithms (although see, e.g., Jacob, Robert and Smith, 2011, for an attempt at this problem); (b) the need to specify a good starting

point for the chain (or alternatively to determine the burn-in period, that is, the length of the initial part of the chain that should be discarded) and (c) the difficulty to assess the convergence of the chain [i.e., to determine if the distribution of $\boldsymbol{\beta}_t$ at iteration t is sufficiently close to the invariant distribution $p(\boldsymbol{\beta}|\mathcal{D})$].

To be fair, these problems are not so critical for binary regression models. Regarding (b), one may simply start the chain from the posterior mode, or from a draw of one of the Gaussian approximations covered in the previous section. Regarding (c) for most standard datasets, MCMC converges reasonably fast, and convergence is easy to assess visually. The main issue in practice is that MCMC generates correlated random variables, and these correlations inflate the Monte Carlo variance.

4.3.1 Gibbs sampling. A well-known MCMC approach to binary regression, due to Albert and Chib (1993), is to apply Gibbs sampling to the joint posterior $p(\boldsymbol{\beta}, \mathbf{z}|\mathcal{D})$ under the data-augmentation formulation (2.2) and (2.3); that is, one iterates the two following steps: (a) sample from $\mathbf{z}|\boldsymbol{\beta}, \mathcal{D}$ and (b) sample from $\boldsymbol{\beta}|\mathbf{z}, \mathcal{D}$.

For (a) and for a probit model, the z_i 's are conditionally independent, and follow a truncated Gaussian distribution

$$p(z_i|\boldsymbol{\beta}, \mathcal{D}) \propto N_1(z_i; \boldsymbol{\beta}^T \mathbf{x}_i, 1) \mathbb{1}\{z_i y_i > 0\},$$

which is easy to sample from (Chopin, 2011). For step (b) and a Gaussian prior $N_p(\mathbf{0}, \boldsymbol{\Sigma}_{\text{prior}})$, one has, thanks to standard conjugacy properties:

$$\boldsymbol{\beta}|\mathbf{z}, \mathcal{D} \sim N_p(\boldsymbol{\Sigma}_{\text{post}}^{-1} \mathbf{xz}, \boldsymbol{\Sigma}_{\text{post}}), \quad \boldsymbol{\Sigma}_{\text{post}}^{-1} = \boldsymbol{\Sigma}_{\text{prior}}^{-1} + \mathbf{x}\mathbf{x}^T,$$

where \mathbf{x} is the $n \times p$ matrix obtained by stacking the \mathbf{x}_i^T . Note that $\boldsymbol{\Sigma}_{\text{post}}$ and its inverse need to be computed only once, hence the complexity of a Gibbs iteration is $\mathcal{O}(p^2)$, not $\mathcal{O}(p^3)$.

The main drawback of Gibbs sampling is that it is particularly not generic: its implementation depends very strongly on the prior and the model. Sticking to the probit case, switching to another prior requires deriving a new way to update $\boldsymbol{\beta}|\mathbf{z}, \mathcal{D}$. For instance, for a prior which is a product of Students with scales σ_j (e.g., our Cauchy prior), one may add extra latent variables, by resorting to the well-known representation: $\beta_j|s_j \sim N_1(0, v\sigma_j^2/s_j)$, $s_j \sim \text{Chi}^2(v)$; with $v = 1$ for our Cauchy prior. Then the algorithm has three steps: (a) an update of the z_i 's, exactly as above; (b) an update of $\boldsymbol{\beta}$, as above but with $\boldsymbol{\Sigma}_{\text{prior}}$ replaced by the diagonal matrix with elements $v\sigma_j^2/s_j$, $j = 1, \dots, p$; and (c) an

(independent) update of the p latent variables s_j , with $s_j|\boldsymbol{\beta}, \mathbf{z}, \mathcal{D} \sim \text{Gamma}((1+\nu)/2, (1+\nu\beta_j^2/\sigma_j^2)/2)$. The complexity of step (b) is now $\mathcal{O}(p^3)$, since $\boldsymbol{\Sigma}_{\text{prior}}$ and $\boldsymbol{\Sigma}_{\text{post}}$ must be recomputed at each iteration (although some speed-up may be obtained by using Sherman–Morrison formula).

Of course, considering yet another type of prior would require deriving another strategy for sampling $\boldsymbol{\beta}$. Then if one turns to logistic regression, things get rather complicated. In fact, deriving an efficient Gibbs sampler for logistic regression is a topic of current research; see Holmes and Held (2006), Frühwirth-Schnatter and Frühwirth (2010), Gramacy and Polson (2012), Polson, Scott and Windle (2013). In a nutshell, the two first papers use the same data augmentation as above, but with $\epsilon_i \sim \text{Logistic}(1)$ written as a certain mixture of Gaussians (infinite for the first paper, finite but approximate for the second paper), while Polson, Scott and Windle (2013) use instead a representation of a logistic likelihood as an infinite mixture of Gaussians, with a Poly–Gamma as the mixing distribution. Each representation leads to introducing extra latent variables, and discussing how to sample their conditional distributions.

Since their implementation is so model-dependent, the main justification for Gibbs samplers should be their greater performance relative to more generic algorithms. We will investigate if this is indeed the case in our numerical section.

4.3.2 Hastings–Metropolis. Algorithm 1 gives a generic description of an iteration of a Hastings–Metropolis algorithm that samples from $p(\boldsymbol{\beta}|\mathcal{D})$. Much like importance sampling, Hastings–Metropolis is both simple and generic, that is, up to the choice of the proposal kernel $\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ (the distribution of the proposed point $\boldsymbol{\beta}^*$, given the current point $\boldsymbol{\beta}$). A naive approach is to take $\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ independent of $\boldsymbol{\beta}$, $\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta}) = q(\boldsymbol{\beta}^*)$, where q is some approximation of the posterior. In

Algorithm 1 Hastings–Metropolis iteration

Input $\boldsymbol{\beta}$

Output $\boldsymbol{\beta}'$

1 Sample $\boldsymbol{\beta}^* \sim \kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta})$.

2 With probability $1 \wedge r$,

$$r = \frac{p(\boldsymbol{\beta}^*)p(\mathcal{D}|\boldsymbol{\beta}^*)\kappa(\boldsymbol{\beta}|\boldsymbol{\beta}^*)}{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta})},$$

set $\boldsymbol{\beta}' \leftarrow \boldsymbol{\beta}^*$; otherwise set $\boldsymbol{\beta}' \leftarrow \boldsymbol{\beta}$.

practice, this usually does not work better than importance sampling based on the same proposal, hence this strategy is hardly used.

A more usual strategy is to set the proposal kernel to a random walk: $\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta}) = N_p(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\text{prop}})$. It is well known that the choice of $\boldsymbol{\Sigma}_{\text{prop}}$ is critical for good performance. For instance, in the univariate case, if $\boldsymbol{\Sigma}_{\text{prop}}$ is too small, the chain moves slowly, while if too large, proposed moves are rarely accepted.

A result from the optimal scaling literature (e.g., Roberts and Rosenthal, 2001) is that, for a $N_p(\mathbf{0}, \mathbf{I}_p)$ target, $\boldsymbol{\Sigma}_{\text{prop}} = (\lambda^2/p)\mathbf{I}_p$ with $\lambda = 2.38$ is asymptotically optimal, in the sense that as $p \rightarrow \infty$, this choice leads to the fastest exploration. Since the posterior of a binary regression model is reasonably close to a Gaussian, we adapt this result by taking $\boldsymbol{\Sigma}_{\text{prop}} = (\lambda^2/p)\boldsymbol{\Sigma}_q$ in our simulations, where $\boldsymbol{\Sigma}_q$ is the covariance matrix of a (Laplace or EP) Gaussian approximation of the posterior. This strategy seems validated by the fact we obtain acceptance rates close to the optimal rate, as given by Roberts and Rosenthal (2001).

The bad news behind this optimality result is that the chain requires $\mathcal{O}(p)$ steps to move a $\mathcal{O}(1)$ distance. Thus, random walk exploration tends to become slow for large p . This is usually cited as the main motivation to develop more elaborate MCMC strategies, such as HMC, which we cover in the following section.

4.3.3 HMC. Hamiltonian Monte Carlo (HMC, also known as Hybrid Monte Carlo, Duane et al., 1987) is a type of MCMC algorithm that performs several steps in the parameter space before determining if the new position is accepted or not. Thus HMC tends to make bigger jumps in the parameter space than standard Hastings–Metropolis. HMC has been known for some time in physics, but it seems to have been considered only recently in statistics, thanks in particular to the excellent review of Neal (2011).

Consider the pair $(\boldsymbol{\beta}, \boldsymbol{\alpha})$, where $\boldsymbol{\beta} \sim p(\boldsymbol{\beta}|\mathcal{D})$, and $\boldsymbol{\alpha} \sim N_p(0, M^{-1})$, thus with joint un-normalised density $\exp\{-H(\boldsymbol{\beta}, \boldsymbol{\alpha})\}$, with

$$H(\boldsymbol{\beta}, \boldsymbol{\alpha}) = E(\boldsymbol{\beta}) + \frac{1}{2}\boldsymbol{\alpha}^T M \boldsymbol{\alpha},$$

$$E(\boldsymbol{\beta}) = -\log\{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\}.$$

The physical interpretation of HMC is that of a particle at position $\boldsymbol{\beta}$, with velocity $\boldsymbol{\alpha}$, potential energy $E(\boldsymbol{\beta})$, kinetic energy $\frac{1}{2}\boldsymbol{\alpha}^T M \boldsymbol{\alpha}$, for some mass matrix M , and therefore total energy given by $H(\boldsymbol{\beta}, \boldsymbol{\alpha})$. The particle is expected to follow a trajectory such that $H(\boldsymbol{\beta}, \boldsymbol{\alpha})$ remains constant over time.

Algorithm 2 Leap-frog step**Input** $(\boldsymbol{\beta}, \boldsymbol{\alpha})$ **Output** $(\boldsymbol{\beta}_1, \boldsymbol{\alpha}_1)$ **1** $\boldsymbol{\alpha}_{1/2} \leftarrow \boldsymbol{\alpha} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\beta}} E(\boldsymbol{\beta})$ **2** $\boldsymbol{\beta}_1 \leftarrow \boldsymbol{\beta} + \epsilon \boldsymbol{\alpha}_{1/2}$ **3** $\boldsymbol{\alpha}_1 \leftarrow \boldsymbol{\alpha}_{1/2} - \frac{\epsilon}{2} \nabla_{\boldsymbol{\beta}} E(\boldsymbol{\beta}_1)$

In practice, HMC proceeds as follows: first, sample a new velocity vector, $\boldsymbol{\alpha} \sim N_p(0, \boldsymbol{M}^{-1})$. Second, move the particle while keeping the Hamiltonian H constant; in practice, discretisation must be used, so L steps of step-size ϵ are performed through leap-frog steps; see Algorithm 2 which describes one such step. Third, the new position, obtained after L leap-frog steps is accepted or rejected according to probability $1 \wedge \exp\{H(\boldsymbol{\beta}, \boldsymbol{\alpha}) - H(\boldsymbol{\beta}^*, \boldsymbol{\alpha}^*)\}$; see Algorithm 3 for a summary. The validity of the algorithm relies on the fact that a leap-frog step is “volume preserving”; that is, the deterministic transformation $(\boldsymbol{\beta}, \boldsymbol{\alpha}) \rightarrow (\boldsymbol{\beta}_1, \boldsymbol{\alpha}_1)$ has Jacobian one. This is why the acceptance probability admits this simple expression.

The tuning parameters of HMC are \boldsymbol{M} (the mass matrix), L (number of leap-frog steps), and ϵ (the step-size). For \boldsymbol{M} , we follow Neal’s (2011) recommendation to take $\boldsymbol{M}^{-1} = \boldsymbol{\Sigma}_q$, an approximation of the posterior variance (again obtained from either Laplace or EP). This is equivalent to rescaling the posterior so as to have a covariance matrix close to identity. In this way, we avoid the bad mixing typically incurred by strong correlations between components.

The difficulty to choose L and ϵ seems to be the main drawback of HMC. The performance of HMC seems very sensitive to these tuning parameters, yet clear guidelines on how to choose them seem currently lacking. A popular approach is to fix $L\epsilon$ to some value, and to use vanishing adaptation (Andrieu and Thoms, 2008) to adapt ϵ so as to target acceptance rate of

0.65 (the optimal rate according to the formal study of HMC by Beskos et al., 2013): that is, at iteration t , take $\epsilon = \epsilon_t$, with $\epsilon_t = \epsilon_{t-1} - \eta_t (R_t - 0.65)$, $\eta_t = t^{-\kappa}$, $\kappa \in (1/2, 1)$ and R_t the acceptance rate up to iteration t . The rationale for fixing $L\epsilon$ is that quantity may be interpreted as a “simulation length,” that is, how much distance one moves at each step; if too small, the algorithm may exhibit random walk behaviour, while if too large, it may move a long distance before coming back close to its starting point. Since the spread is already taken into account through $\boldsymbol{M}^{-1} = \boldsymbol{\Sigma}_q$, we took $\epsilon L = 1$ in our simulations.

4.3.4 NUTS and other variants of HMC. Girolami and Calderhead (2011) proposed an interesting variation of HMC, where the mass matrix \boldsymbol{M} is allowed to depend on $\boldsymbol{\beta}$; for example, $\boldsymbol{M}(\boldsymbol{\beta})$ is set to the Fisher information of the model. This allows the corresponding algorithm, called RHMC (Riemannian HMC), to adapt locally to the geometry of the target distribution. The main drawback of RHMC is that each iteration involves computing derivatives of $\boldsymbol{M}(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, which is very expensive, especially if p is large. For binary regression, we found RMHC to be too expensive relative to plain HMC, even when taking into account the better exploration brought by RHMC. This might be related to the fact that the posterior of a binary regression model is rather Gaussian-like, and thus may not require such a local adaptation of the sampler.

We now focus on NUTS (No U-Turn sampler, Hoffman and Gelman, 2014), a variant of HMC which does not require the user to specify a priori L , the number of leap-frog steps. Instead, NUTS aims at keeping on doing such steps until the trajectory starts to loop back to its initial position. Of course, the difficulty in this exercise is to preserve the time reversibility of the simulated Markov chain. To that effect, NUTS constructs iteratively a binary tree whose leaves correspond to different velocity-position pairs $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ obtained after a certain number of leap-frog steps. The tree starts with two leaves, one at the current velocity-position pair, and another leaf that corresponds to one leap-frog step, either in the forward or backward direction (i.e., by reversing the sign of velocity); then it iteratively doubles the number of leaves, by taking twice more leap frog steps, again either in the forward or backward direction. The tree stops growing when at least one leaf corresponds to a “U-turn”; then NUTS chooses randomly one leaf, among those leaves that would have generated the current position with the same binary tree mechanism; in this way reversibility

Algorithm 3 HMC iteration**Input** $\boldsymbol{\beta}$ **Output** $\boldsymbol{\beta}'$ **1** Sample momentum $\boldsymbol{\alpha} \sim N_p(0, \boldsymbol{M})$.**2** Perform L leap-frog steps (see Algorithm 2), starting from $(\boldsymbol{\beta}, \boldsymbol{\alpha})$; call $(\boldsymbol{\beta}^*, \boldsymbol{\alpha}^*)$ the final position.**3** With probability $1 \wedge r$,

$$r = \exp\{H(\boldsymbol{\beta}, \boldsymbol{\alpha}) - H(\boldsymbol{\beta}^*, \boldsymbol{\alpha}^*)\}$$

set $\boldsymbol{\beta}' = \boldsymbol{\beta}^*$; otherwise set $\boldsymbol{\beta}' = \boldsymbol{\beta}$.

is preserved. Finally, NUTS proposes the new position that corresponds to the chosen leaf.

We refer the readers to [Hoffman and Gelman \(2014\)](#) for a more precise description of NUTS. Given its complexity, implementing directly NUTS seems to require more efforts than the other algorithms covered in this paper. Fortunately, the STAN package (<http://mc-stan.org/>) provides a C++ implementation of NUTS which is both efficient and user-friendly: the only required input is a description of the model in a probabilistic programming language similar to BUGS. In particular, STAN is able to automatically derive the log-likelihood and its gradient, and no tuning of any sort is required from the user. Thus, we will use STAN to assess NUTS in our numerical comparisons.

4.4 Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a class of algorithms for approximating iteratively a sequence of distributions π_t , $t = 0, \dots, T$, using importance sampling, resampling and MCMC steps. We focus here on the nonsequential use of SMC ([Neal, 2001](#), [Chopin, 2002](#), [Del Moral, Doucet and Jasra, 2006](#)), where one is only interested in approximating the final distribution π_T [in our case, the posterior $p(\boldsymbol{\beta}|\mathcal{D})$], and the previous π_t 's are designed so as to allow for a smooth progression from some π_0 , which is easy to sample from, to π_T .

At iteration t , SMC produces a set of weighted particles (simulations) $(\boldsymbol{\beta}_n, w_n)_{n=1}^N$ that approximates π_t , in the sense that

$$\frac{1}{\sum_{n=1}^N w_n} \sum_{n=1}^N w_n \varphi(\boldsymbol{\beta}_n) \rightarrow \mathbb{E}^{\pi_t}[\varphi(\boldsymbol{\beta})]$$

as $N \rightarrow +\infty$. At time 0, one samples $\boldsymbol{\beta}^0 \sim \pi_0$, and set $w_n \leftarrow 1$. To progress from π_{t-1} to π_t , one uses importance sampling: weights are multiplied by ratio $\pi_t(\boldsymbol{\beta}_n)/\pi_{t-1}(\boldsymbol{\beta}_n)$. When the variance of the weights gets too large (which indicates that too few particles contribute significantly to the current approximation), one resamples the particles: each particle gets reproduced O_n times, where $O_n \geq 0$ is random, and such that $\mathbb{E}(O_n) = N w_n / \sum_{m=1}^N w_m$, and $\sum_{n=1}^N O_n = N$ with probability one. In this way, particles with low weights are likely to die, while particles with a large weight get reproduced many times. [In our numerical study, we use specifically the systematic resampling algorithm of [Carpenter, Clifford and Fearnhead, 1999](#); see the supplemental article [Chopin and Ridgway, 2016](#) for an algorithmic description.] Finally, one may reintroduce diversity among the particles by applying one (or several) MCMC steps, using

Algorithm 4 Tempering SMC

Operations involving index n must be performed for all $n \in 1 : N$.

0 Sample $\boldsymbol{\beta}_n \sim q(\boldsymbol{\beta})$ and set $\underline{\delta} \leftarrow 0$.

1 Let, for $\delta \in [\underline{\delta}, 1]$,

$$\text{EF}(\delta) = \frac{1}{N} \frac{\{\sum_{n=1}^N u_\delta(\boldsymbol{\beta}_n)\}^2}{\{\sum_{n=1}^N u_\delta(\boldsymbol{\beta}_n)^2\}},$$

$$u_\delta(\boldsymbol{\beta}) = \left\{ \frac{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})}{q(\boldsymbol{\beta})} \right\}^\delta.$$

If $\text{EF}(1) \geq \tau$, stop and return $(\boldsymbol{\beta}_n, w_n)_{n=1:N}$ with $w_n = u_1(\boldsymbol{\beta}_n)$; otherwise, use the bisection method ([Press et al., 2007](#), Chapter 9) to solve numerically in δ the equation $\text{EF}(\gamma) = \tau$.

2 Resample according to normalised weights $W_n = w_n / \sum_{m=1}^N w_m$, with $w_n = u_\delta(\boldsymbol{\beta}_n)$; see the supplement [Chopin and Ridgway \(2016\)](#) for a description of particular resampling algorithm (known as systematic resampling).

3 Update the $\boldsymbol{\beta}_n$'s through m MCMC steps that leaves invariant $\pi_t(\boldsymbol{\beta})$, using, for example, Algorithm 1 with $\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta}) = N_p(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\text{prop}})$, $\boldsymbol{\Sigma}_{\text{prop}} = \lambda \hat{\boldsymbol{\Sigma}}$, where $\hat{\boldsymbol{\Sigma}}$ is the empirical covariance matrix of the resampled particles.

4 Set $\underline{\delta} \leftarrow \delta$. Go to step 1.

a MCMC kernel that leaves invariant the current distribution π_t .

We focus in this paper on tempering SMC, where the sequence

$$\pi_t(\boldsymbol{\beta}) \propto q(\boldsymbol{\beta})^{1-\delta_t} \{p(\boldsymbol{\beta})p(\mathcal{D}|\boldsymbol{\beta})\}^{\delta_t}$$

corresponds to a linear interpolation (on the log-scale) between some distribution $\pi_0 = q$, and $\pi_T(\boldsymbol{\beta}) = p(\boldsymbol{\beta}|\mathcal{D})$, our posterior. This is a convenient choice in our case, as we have at our disposal some good approximation q (either from Laplace or EP) of our posterior. A second advantage of tempering SMC is that one can automatically adapt the ‘‘temperature ladder’’ δ_t ([Jasra et al., 2011](#)). Algorithm 4 describes a tempering SMC algorithm based on such an adaptation scheme: at each iteration, the next distribution π_t is chosen so that the efficiency factor (defined in Section 4.1) of the importance sampling step from π_{t-1} to π_t equals a predefined level $\tau \in (0, 1)$; a default value is $\tau = 1/2$.

Another part of Algorithm 4 which is easily amenable to automatic calibration is the MCMC step.

We use a random walk Metropolis step, that is, Algorithm 1 with proposal kernel $\kappa(\boldsymbol{\beta}^*|\boldsymbol{\beta}) = N_p(\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\text{prop}})$, but with $\boldsymbol{\Sigma}_{\text{prop}}$ calibrated to the empirical variance of the particles $\hat{\boldsymbol{\Sigma}}: \boldsymbol{\Sigma}_{\text{prop}} = \lambda \hat{\boldsymbol{\Sigma}}$, for some λ . Finally, one may also automatically calibrate the number m of MCMC steps, as in Ridgway (2016), but in our simulations we simply took $m = 3$.

In the end, one obtains essentially a black-box algorithm. In practice, we shall often observe that, for simple datasets, our SMC algorithm automatically reduces to a single importance sampling step, because the efficiency factor of moving from the initial distribution q to the posterior is high enough. In that case, our SMC sampler performs exactly as standard importance sampling.

Finally, we note that the reweighting step and the MCMC steps of Algorithm 4 are easy to parallelise.

5. NUMERICAL STUDY

The point of this section is to compare numerically the different methods discussed in the previous sections, first on several datasets of standard size (that are representative of previous numerical studies), then in a second time on several bigger datasets.

We focus on the following quantities: the marginal likelihood of the data, $p(\mathcal{D})$, and the p marginal posterior distributions of the regression coefficients β_j . Regarding the latter, we follow Faes, Ormerod and Wand (2011) in defining the ‘‘marginal accuracy’’ of approximation q for component j to be

$$\text{MA}_j = 1 - \frac{1}{2} \int_{-\infty}^{+\infty} |q(\beta_j) - p(\beta_j|\mathcal{D})| d\beta_j.$$

This quantity lies in $[0, 1]$, and is scale-invariant. Since the true marginals $p(\beta_j|\mathcal{D})$ are not available, we will approximate them through a Gibbs sampler run for a very long time. To give some scale to this criterion, assume $q(\beta_j) = N_1(\beta_j; \mu_1, \sigma^2)$, $p(\beta_j|\mathcal{D}) = N_1(\beta_j; \mu_2, \sigma^2)$, then MA_j is $2\Phi(-\delta/2) \approx 1 - 0.4 \times \delta$ for $\delta = |\mu_1 - \mu_2|/\sigma$ small enough; for example, 0.996 for $\delta \approx 0.01$, 0.96 for $\delta \approx 0.1$.

In our results, we will refer to the following four prior/model ‘‘scenarios’’: Gaussian/probit, Gaussian/logit, Cauchy/probit, Cauchy/logit, where Gaussian and Cauchy refer to the two priors discussed in Section 2.1. All the algorithms have been implemented in C++, using the Armadillo and Boost libraries, and run on a standard desktop computer (except when explicitly stated). Results for NUTS were obtained by running STAN (<http://mc-stan.org/>) version 2.4.0.

TABLE 1

Datasets of moderate size [from UCI repository, except Elections, from the website of Gelman and Hill’s (2006) book]: name (short and long version), number of instances $n_{\mathcal{D}}$, number of covariates p (including an intercept)

Dataset	$n_{\mathcal{D}}$	p
Pima (Indian diabetes)	532	8
German (credit)	999	25
Heart (Statlog)	270	14
Breast (cancer)	683	10
Liver (Indian Liver patient)	579	11
Plasma (blood screening data)	32	3
Australian (credit)	690	15
Elections	2015	52

5.1 Datasets of Moderate Size

Table 1 lists the 7 datasets considered in this section [obtained from the UCI machine learning repository, except Pima, which is the version available in the R package MASS, and Elections, which is available on the web page of Gelman and Hill’s (2006) book]. These datasets are representative of the numerical studies found in the literature. In fact, it is a super-set of the real datasets considered in Girolami and Calderhead (2011), Shahbaba et al. (2014), Holmes and Held (2006) and also (up to one dataset with 5 covariates) Polson, Scott and Windle (2013). In each case, an intercept has been included; that is, p is the number of predictors plus one.

5.1.1 Fast approximations. We compare the four approximation schemes described in Section 3: Laplace, Improved Laplace, Laplace EM, and EP. We concentrate on the Cauchy/logit scenario for two reasons: (i) Laplace EM requires a Student prior; and (ii) Cauchy/logit seems to be the most challenging scenario for EP, as (a) a Cauchy prior is more difficult to deal with than a Gaussian prior in EP; and (b) contrary to the probit case, the site update requires some approximation; see Section 3.5 for more details.

Left panel of Figure 1 plots the marginal accuracies of the four approximation schemes across all components and all datasets; Figure 2 does the same, but separately for four selected datasets; results for the remaining datasets are available in the supplement (Chopin and Ridgway, 2016).

EP seems to be the most accurate method on these datasets: marginal accuracy is about 0.99 across all components for EP, while marginal accuracy of the other approximation schemes tend to be lower, and

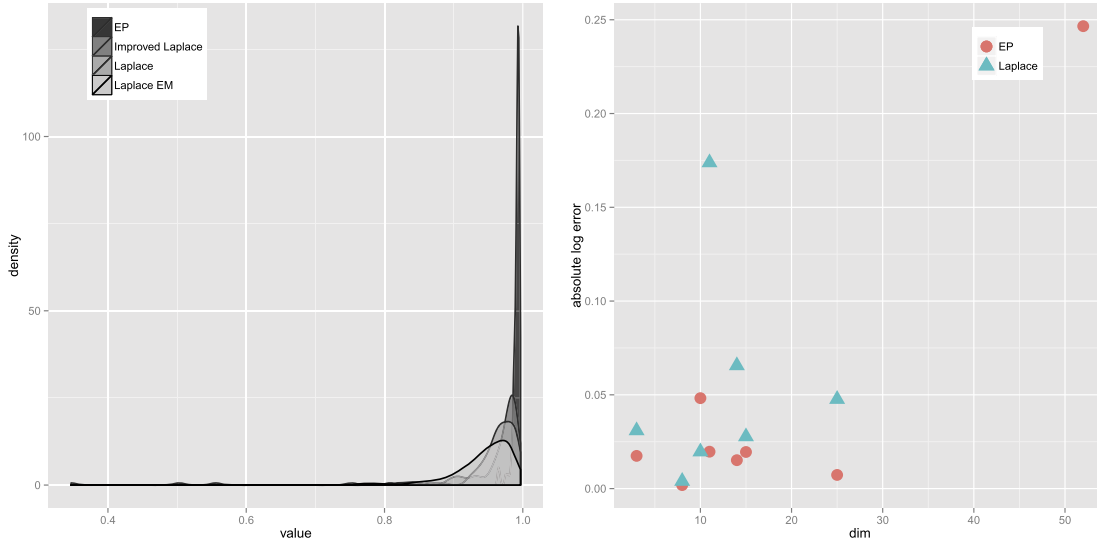


FIG. 1. Comparison of approximation schemes across all datasets of moderate size: marginal accuracies (left), and absolute error for log-evidence versus the dimension p (right); x -axis range of the left plot determined by range of marginal accuracies (i.e., marginal accuracy may drop below 0.4, for example, Laplace-EM).

may even drop to quite small values; see, for example, the German dataset, and the left tail in the left panel of Figure 1.

EP also fared well in terms of CPU time: it was at most seven times as intensive as standard Laplace across the considered datasets, and about 10 to 20 times faster than Improved Laplace and Laplace EM. As expected (see Section 3.6). Of course, the usual caveats apply regarding CPU time comparison, and how they may depend on the hardware, the implementation, and so on.

We also note in passing the disappointing performance of Laplace EM, which was supposed to replace standard Laplace when the prior is Student, but which actually performs not as well as standard Laplace on these datasets.

We refer the reader to the supplement (Chopin and Ridgway, 2016) for similar results on the three other scenarios, which are consistent with those above. In addition, we also represent the approximation error of EP and Laplace for approximating the log-evidence in the right panel of Figure 1. Again, EP is found to be more accurate than Laplace for most datasets (except for the Breast dataset).

To conclude, it seems that EP may safely be used as a complete replacement of sampling-based methods on such datasets, as it produces nearly instant results, and the approximation error along all dimensions is essentially negligible.

5.1.2 Importance sampling, QMC. We now turn to importance sampling (IS), which we deemed our “gold standard” among sampling-based methods, because of its ease of use and other nice properties as discussed in Section 4.1. We use $N = 5 \times 10^5$ samples, and a Gaussian EP proposal. (Results with a Laplace proposal are roughly similar.) We consider first the Gaussian/probit scenario, because this is particularly favourable to Gibbs sampling; see the next section. Table 2 reports for each dataset the efficiency factor of IS (as defined in Section 4.1), the CPU time and two other quantities discussed below.

We see that all these efficiency factors are all close to one, which means IS works almost as well as IID sampling would on such datasets. Further improvement may be obtained by using either parallelization, or QMC (Quasi-Monte Carlo, see Section 4.2). Table 2 reports the speed-up factor obtained when implementing multi-threading on our desktop computer which has a multi threading quad core CPU (hence 8 virtual cores). We also implemented IS on an Amazon EC2 instance with 32 virtual CPUs, and obtained speed-up factors about 20, and running times below 2s.

Finally, Table 2 also reports the MSE improvement (i.e., MSE ratio of IS relative to IS-QMC) obtained by using QMC, or more precisely RQMC (randomised QMC), based on a scrambled Sobol’ sequence (see, e.g., Lemieux, 2009). Specifically, the table reports the median MSE improvement for the p posterior expectations (first column), and the MSE improvement for the

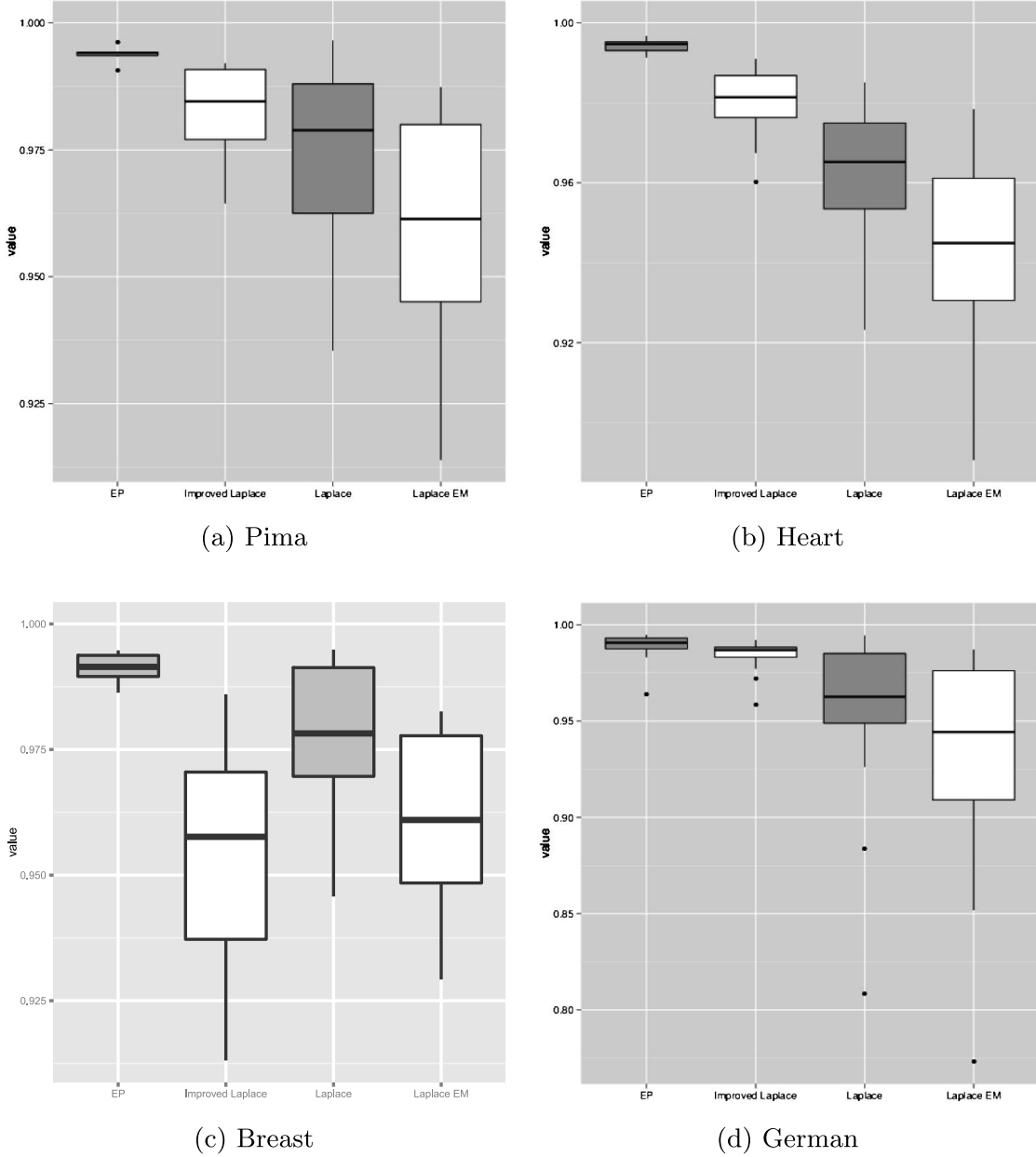


FIG. 2. Box-plots of marginal accuracies across the p dimensions, for the four approximation schemes, and four selected datasets; plots for remaining datasets are in the supplement (Chopin and Ridgway, 2016). For the sake of readability, scale of y -axis varies across plots.

evidence (second column). The improvement brought by RQMC varies strongly across datasets.

The efficiency gains brought by parallelization and QMC may be combined, because the bulk of the computation (as reported by a profiler) is the N likelihood evaluations, which are trivial to parallelize.

It is already clear that other sampling-based methods do not really have a fighting chance on such datasets, but we shall compare them in the next section for the sake of completeness. See also the supplement (Chopin

and Ridgway, 2016) for results for other scenarios, which are very much in line with those above.

5.1.3 MCMC schemes. In order to compare the different sampling-based methods, we define the IRIS (Inefficiency Relative to Importance Sampling) criterion, for a given method M and a given posterior estimate, as follows:

$$\frac{\text{MSE}_M}{\text{MSE}_{\text{IS}}} \times \frac{\text{CPU}_{\text{IS}}}{\text{CPU}_M},$$

TABLE 2

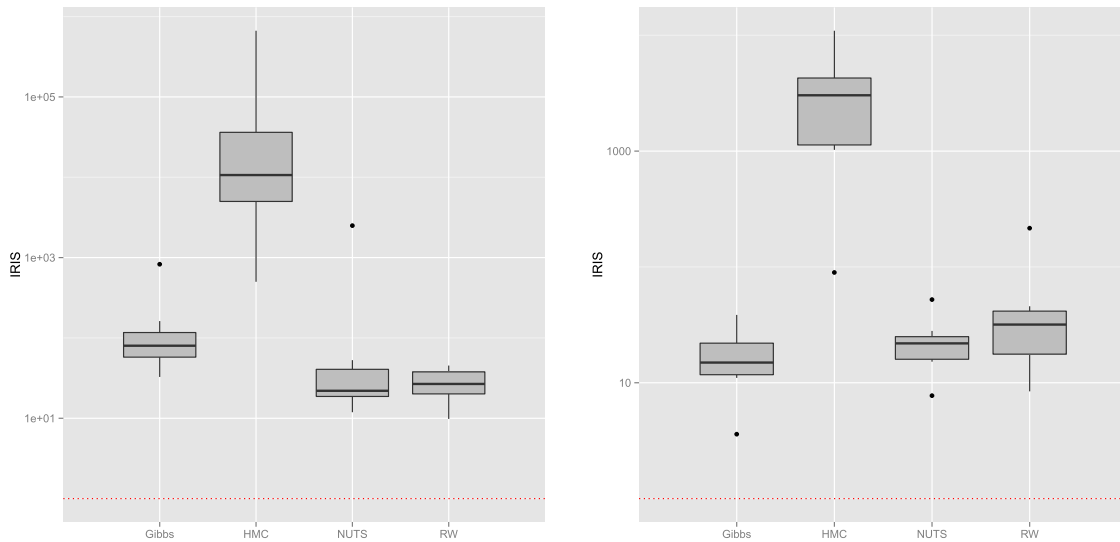
Performance of importance sampling (IS), and QMC importance sampling (IS-QMC), on all datasets, in Gaussian/probit scenario: efficiency factor (EF), CPU time (in seconds), speed gain when using multi-threading Intel hyper-threaded quad core CPU (Speed gain MT), and efficiency gain of QMC (see text)

Dataset	IS			IS-QMC	
	EF = ESS / N	CPU time	MT speed-up	MSE improv. (expectation)	MSE improv. (evidence)
Pima	99.5%	37.54 s	4.39	28.9	42.7
German	97.9%	79.65 s	4.51	13.2	8.2
Breast	82.9%	50.91 s	4.45	2.6	6.2
Heart	95.2%	22.34 s	4.53	8.8	9.3
Liver	74.2%	35.93 s	4.76	7.6	11.3
Plasma	90.0%	2.32 s	4.28	2.2	4.4
Australian	95.6%	53.32 s	4.57	12	20.3
Elections	21.39%	139.48 s	3.87	617.9	3.53

where MSE_M (resp., MSE_{IS}) is the mean square error of the posterior estimate obtained with method M (resp., with importance sampling), and CPU_M the CPU time of method M (resp., importance sampling). The comparison is relative to importance sampling *without* parallelisation or quasi-Monte Carlo sampling. In terms of posterior estimates, we consider the expectation and variance of each posterior marginal $p(\beta_j|\mathcal{D})$. We observe that, in both cases, IRIS does not vary much across the p components, so we simply report the median of these p values. Figure 3 reports the me-

dian IRIS across all datasets. We refer the reader to Section 4.3 for how we tuned these MCMC algorithms.

The first observation is that all these MCMC schemes are significantly less efficient than importance sampling on such datasets. The source of inefficiency seems mostly due to the autocorrelations of the simulated chains (for Gibbs or random walk Metropolis), or, equivalently, the number of leap-frog steps performed at each iteration in HMC and NUTS. See the supplement (Chopin and Ridgway, 2016) for ACF's (Autocorrelation plots) to support this statement.



(a) Median IRIS for the p posterior expectations $\mathbb{E}[\beta_j|\mathcal{D}]$ (b) Median IRIS for the p posterior variances $\text{Var}[\beta_j|\mathcal{D}]$

FIG. 3. IRIS (Inefficiency relative to importance sampling) across all datasets for MCMC schemes and Gaussian/probit scenario; left (resp., right) panel shows median IRIS when estimating the p posterior expectations (resp., the p posterior variances).

Second, HMC and NUTS do not perform significantly better than random-walk Metropolis. As already discussed, HMC-type algorithms are expected to outperform random walk algorithms as $p \rightarrow +\infty$. But the considered datasets seem too small to give evidence to this phenomenon, and should not be considered as reasonable benchmarks for HMC-type algorithms (not to mention again that these algorithms are significantly outperformed by IS on such datasets). We note in passing that it might be possible to get better performance for HMC by finely tuning the quantities ϵ and L on per dataset basis. We have already explained in the [Introduction](#) why we think this is bad practice, and we also add at this stage that the fact HMC requires so much more effort to obtain good performance (relative to other MCMC samplers) is a clear drawback.

Regarding Gibbs sampling, it seems a bit astonishing that an algorithm specialised to probit regression is not able to perform better than more generic approach on such simple datasets. Recall that the Gaussian/probit case is particularly favourable to Gibbs, as explained in [Section 4.3.1](#). See the supplement ([Chopin and Ridgway, 2016](#)) for a comparison of MCMC schemes in other scenarios than Gaussian/probit; results are roughly similar, except that Gibbs is more significantly outperformed by other methods, as expected.

5.2 Bigger Datasets

Finally, we turn our attention to the bigger datasets summarised by [Table 3](#). These datasets not only have more covariates (than those of the previous section), but also stronger correlations between these covariates (especially Sonar and Musk). We consider the probit/Gaussian scenario.

Regarding fast approximations, we observe again that EP performs very well, and better than Laplace; see [Figure 4](#). It is only for DNA (180 covariates) that the EP approximation starts to suffer.

Regarding sampling-based methods, importance sampling may no longer be used as a reference, as the effective sample size collapses to a very small

value for these datasets. We replace it by the tempering SMC algorithm described in [Section 4.4](#). Moreover, we did not manage to calibrate HMC so as to obtain reasonable performance in this setting. Thus, among sampling-based algorithms, the four remaining contenders are: Gibbs sampling, NUTS, RWHM (random walk Hastings–Metropolis), and tempering SMC. Recall that the last two are calibrated with the approximation provided by EP.

[Figure 5](#) reports the “effective sample size” of the output of these algorithms when run for the same fixed CPU time (corresponding to 5×10^5 iterations of RWHM), for the p posterior expectations (left panels), and the p posterior variances (right panels); here “effective sample size” is simply the posterior variance divided by the MSE of the estimate (across 50 independent runs of the same algorithm).

No algorithm seems to vastly outperform the others consistently across the three datasets. If anything, RWHM seems to show consistently best or second best performance.

Still, these results offer the following insights. Again, we see that Gibbs sampling, despite being a specialised algorithm, does not outperform significantly more generic algorithms. Recall that the probit/Gaussian scenario is very favourable to Gibbs sampling; in other scenarios (results not shown), Gibbs is strongly dominated by other algorithms.

More surprisingly, RWHM still performs well despite the high dimension. In addition, RWHM seems more robust than SMC to an imperfect calibration; see the DNA example, where the error of the EP approximation is greater.

On the other hand, SMC is more amenable to parallelisation, hence on a parallel architecture, SMC would be likely to outperform the other approaches.

6. VARIABLE SELECTION

We discuss in this section the implications of our findings for variable selection. The standard way to formalise variable selection is to introduce as a parameter the binary vector $\boldsymbol{\gamma} \in \{0, 1\}^p$, and to define the likelihood

$$p(\mathcal{D}|\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{i=1}^{n_{\mathcal{D}}} F(y_i \boldsymbol{\beta}_{\boldsymbol{\gamma}}^T \boldsymbol{x}_{\boldsymbol{\gamma}, i}),$$

where $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ (resp., $\boldsymbol{x}_{\boldsymbol{\gamma}, i}$) is the vector of length $|\boldsymbol{\gamma}|$ that one obtains by excluding from $\boldsymbol{\beta}$ (resp., \boldsymbol{x}_i) the components j such that $\gamma_j = 0$. Several priors may be considered for this problem ([Chipman, George and McCulloch, 2001](#)), but for simplicity, we will assume a uniform prior for $\boldsymbol{\gamma}$ (with respect to the set $\{0, 1\}^p$), and

TABLE 3

Datasets of larger size (from UCI repository): name, number of instances $n_{\mathcal{D}}$, number of covariates p (including an intercept)

Dataset	$n_{\mathcal{D}}$	p
Musk	476	95
Sonar	208	61
DNA	400	180

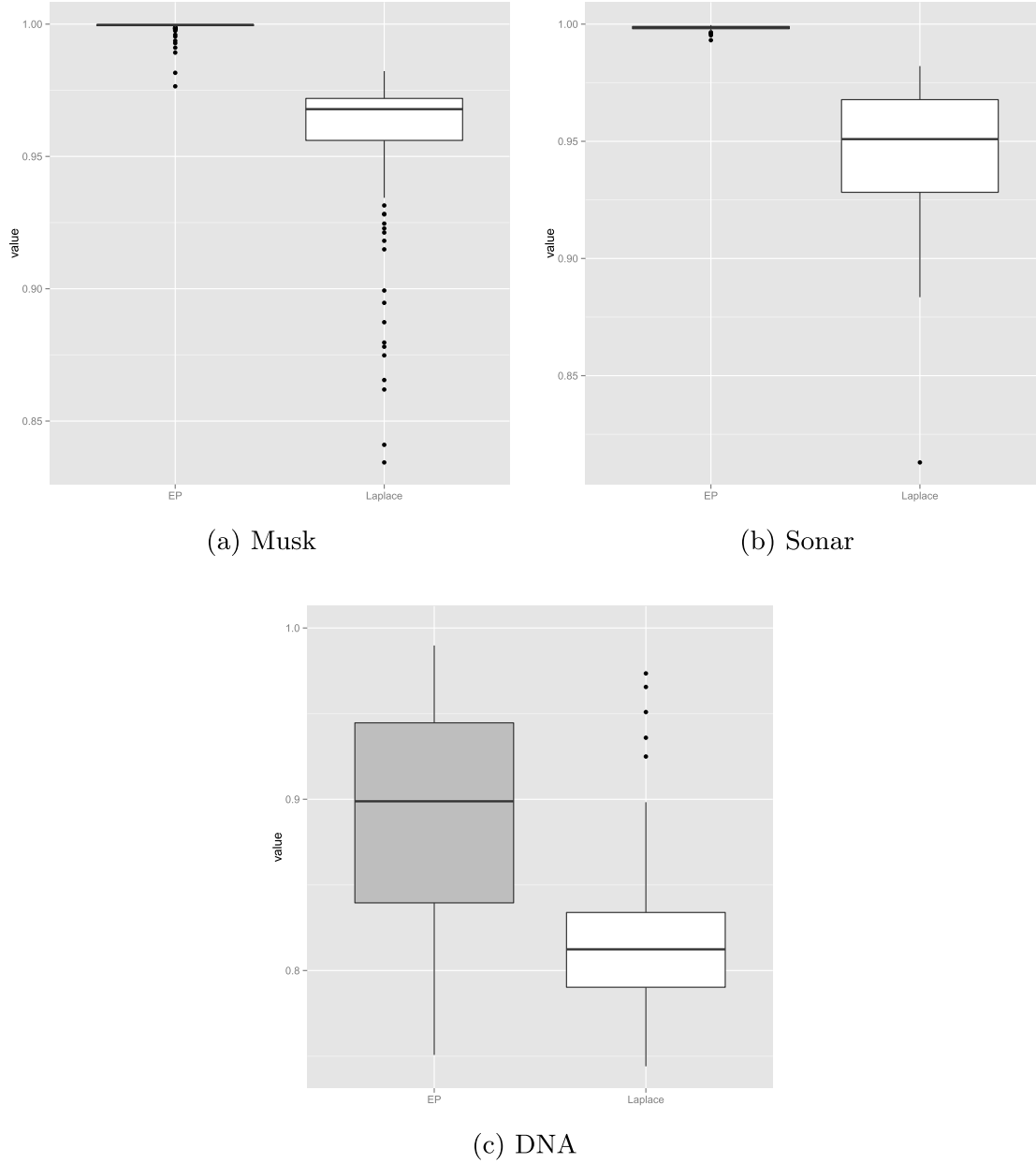


FIG. 4. Marginal accuracies across the p dimensions of EP and Laplace, for datasets Musk, Sonar and DNA.

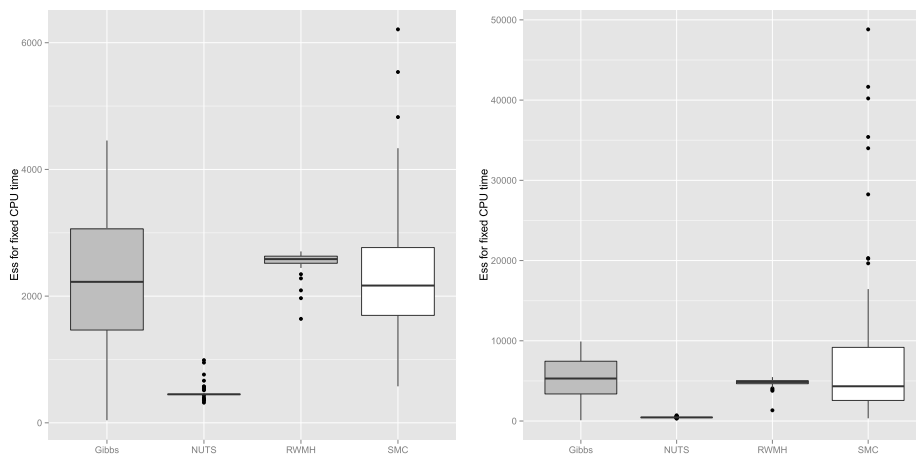
one of the two priors discussed in Section 2.1 for $\beta_{\mathcal{Y}}$, given \mathcal{Y} .

Computationally, variable selection is more challenging than parameter estimation, because the posterior $p(\beta, \mathcal{Y}|\mathcal{D})$ is a mixture of discrete and continuous components. The quantity $p(\mathcal{D}|\mathcal{Y})$ is the marginal likelihood of the data for a given set of predictors. We have seen in the previous section that this quantity is easy to approximate (using, e.g., Laplace, EP, or importance sampling). This suggests to sample directly the marginal posterior distribution $p(\mathcal{Y}|\mathcal{D}) \propto p(\mathcal{Y})p(\mathcal{D}|\mathcal{Y})$. if p is small, one may perform a com-

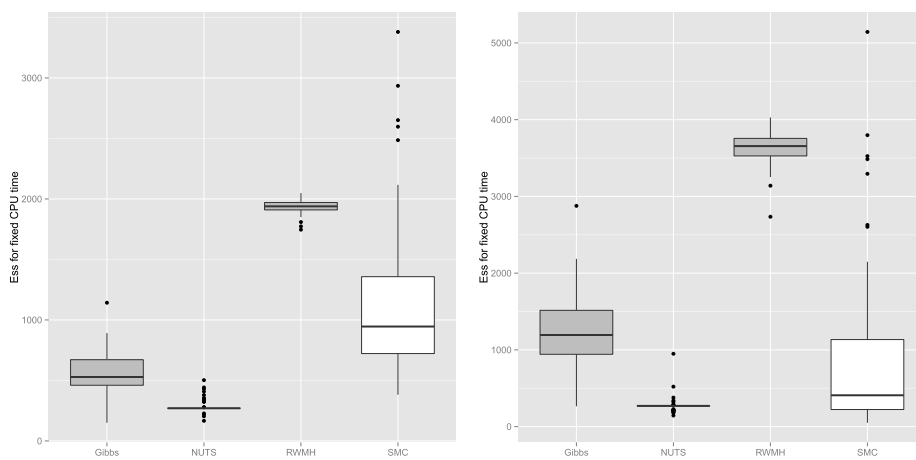
plete enumeration: approximate $p(\mathcal{Y})p(\mathcal{D}|\mathcal{Y})$ for each \mathcal{Y} , and normalise. For larger p , one may adapt the approach of Schäfer and Chopin (2013) for sampling binary vectors, as described in the next sections.

6.1 SMC Algorithm of Schäfer and Chopin (2013)

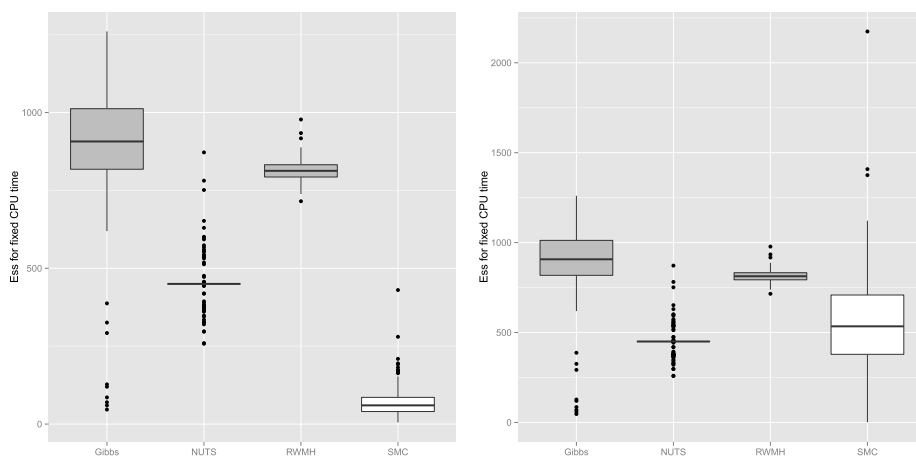
In linear regression, $y_i = \beta_{\mathcal{Y}}^T \mathbf{x}_{\mathcal{Y},i} + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}_1(0, \sigma^2)$, the marginal likelihood $p(\mathcal{D}|\mathcal{Y})$ may be computed exactly (for a certain class of priors). Schäfer and Chopin (2013) use this property to construct a tempering SMC sampler, which transitions from the prior $p(\mathcal{Y})$ to the posterior $p(\mathcal{Y}|\mathcal{D})$, through the tem-



(a) Musk



(b) Sonar



(c) DNA

FIG. 5. *Effective sample size for a fixed CPU time for sampling-based algorithms: posterior expectations (left), and posterior variances (right) for datasets (from top to bottom): Musk, Sonar and ADN.*

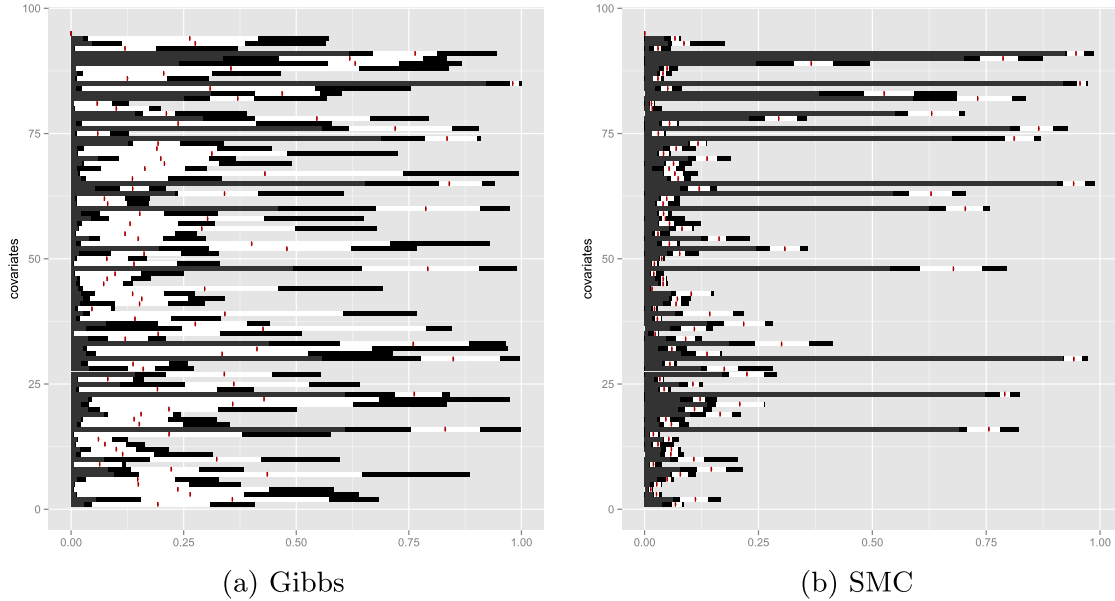


FIG. 6. Variation of estimated inclusion probabilities $p(\gamma_j = 1|\mathcal{D})$ over 50 runs for the p covariates of Musk dataset: median (red line), 80% confidence interval (white box); the black-box extends until the maximum value.

pering sequence $\pi_t(\boldsymbol{\gamma}) \propto p(\boldsymbol{\gamma})p(\mathcal{D}|\boldsymbol{\gamma})^{\delta_t}$, with δ_t growing from 0 to 1. This algorithm has the same structure as Algorithm 4 (with the obvious replacements of the $\boldsymbol{\beta}$'s by $\boldsymbol{\gamma}$'s and so on). The only difference is the MCMC step used to diversify the particles after resampling. Instead of a random walk step (which would be ill-defined on a discrete space), Schäfer and Chopin (2013) use a Metropolis step based on an independent proposal, constructed from a sequence of nested logistic regressions: proposal for first component γ_1 is Bernoulli, proposal for second component γ_2 , conditional on γ_1 , corresponds to a logistic regression with γ_1 and an intercept as covariates, and so on. The parameters of these p successive regressions are simply estimated from the current particle system. Schäfer and Chopin (2013) show that their algorithm significantly outperforms several MCMC samplers on datasets with more than 100 covariates.

6.2 Adaptation to Binary Regression

For binary regression models, $p(\mathcal{D}|\boldsymbol{\gamma})$ is intractable, so the approach of Schäfer and Chopin (2013) cannot be applied directly. On the other hand, we have seen that (a) both Laplace and EP may provide a fast approximation of the evidence $p(\mathcal{D}|\boldsymbol{\gamma})$; and (b) both importance sampling and the tempering SMC algorithm may provide an unbiased estimator of $p(\mathcal{D}|\boldsymbol{\gamma})$.

Based on these remarks, Schäfer (2012) in his Ph.D. thesis considered the following extension of the SMC

algorithm of Schäfer and Chopin (2013): in the sequence $\pi_t(\boldsymbol{\gamma}) \propto p(\boldsymbol{\gamma})p(\mathcal{D}|\boldsymbol{\gamma})^{\delta_t}$, the intractable quantity $p(\mathcal{D}|\boldsymbol{\gamma})$ is simply replaced by an unbiased estimator (obtained with importance sampling and the Gaussian proposal corresponding to Laplace). The corresponding algorithm remains valid, thanks to pseudo-marginal arguments (see, e.g., Andrieu and Roberts, 2009). Specifically, one may reinterpret the resulting algorithm as a SMC algorithm for a sequence of distributions on an extended space, such that marginal in $\boldsymbol{\gamma}$ is exactly the posterior $p(\mathcal{D}|\boldsymbol{\gamma})$ at time $t = T$. In fact, it may be seen as a particular variant of the SMC² algorithm of Chopin, Jacob and Papaspiliopoulos (2013).

6.3 Numerical Illustration

We now compare the proposed SMC approach with the Gibbs sampler of Holmes and Held (2006) for sampling from $p(\boldsymbol{\beta}, \boldsymbol{\gamma}|\mathcal{D})$, on the Musk dataset. Both algorithms were given the same CPU budget (15 minutes), and were run 50 times; see Figure 6. Clearly, the SMC sampler provides more reliable estimates of the inclusion probabilities $p(\gamma_j = 1|\mathcal{D})$ on such a big dataset. See also the Ph.D. dissertation of Schäfer (2012) for results consistent with those, on other datasets, and when comparing to the adaptive reversible jump sampler of Lamnisos, Griffin and Steel (2013).

6.4 Spike and Slab

We also note in passing that a different approach to the variable selection problem is to assign a spike and

slab prior to β (George and McCulloch, 1993):

$$p(\beta) = \prod_{j=1}^p \{ \lambda N_1(\beta_j; 0, v_0^2) + (1 - \lambda) N_1(\beta_j; 0, v_1^2) \}, \quad v_0^2 \ll v_1^2,$$

where $\lambda \in (0, 1)$, v_0^2 and v_1^2 are fixed hyper-parameters. This prior yields a continuous posterior (without point masses at $\beta_j = 0$), which should be easier to sample from than the discrete-continuous mixture obtained in the standard formulation of Bayesian variable selection. It would be interesting to see to which extent our discussion and findings extend to this particular type of prior; see for particular Hernández-Lobato, Hernández-Lobato and Dupont (2013) for an interesting EP algorithm, based on the parametric family (for the sites q_i) corresponding to an independent product of p factors, equal to a Bernoulli (for choosing spike or slab) times a Gaussian (for the coefficient β_j). Unfortunately, we managed to make this approach work for probit regression only on our list of small datasets, but not on, for example, Musk.

7. CONCLUSION

The conclusions we draw and the recommendations we give in this section apply exclusively to binary regression models. Discussing to which extent these conclusions may be generalised to other models is deferred to next section.

7.1 Recommendation to End Users

Our first and perhaps most important message to end users is that Bayesian computation (for binary regression) is now sufficiently fast for routine use: if the right approach is used, results may be obtained near instantly on a standard computer, at least on simple datasets.

Concretely, as far as binary regression is concerned, our main recommendation is to use EP. It is very fast, and its approximation error is negligible even on the big datasets we have considered. EP requires some expertise to implement, but the second author has released a R package (available at <https://cran.r-project.org/web/packages/EPGLM/>) that computes the EP approximation for a logit or a probit model.

In case one wishes to assess the EP error, by running in a second step some exact algorithm, we would recommend to use the SMC approach outlined in Section 4.4 (i.e., with initial particles simulated from the EP approximation). Often, this SMC sampler will reduce to a single importance sampling step, and will

perform extremely well. Even when it does not, it should provide decent performance, especially if run on (and implemented for) a parallel architecture. Alternatively, on a single-core machine, random walk Metropolis is particularly simple to implement, and performs surprisingly well on high-dimensional data (when properly calibrated with EP).

7.2 Recommendations to Experts: Which Benchmark?

The question “Which benchmark to use?” may be recast into (as nicely put by one reviewer) the more general question: what is best practice in demonstrating that a newly proposed method is useful? To answer this question, we shall distinguish between specialised algorithms and generic algorithms.

7.2.1 Benchmarking specialised algorithms. By “specialised algorithm,” we mean in particular Gibbs samplers that are specifically derived for a given binary regression model, and a given prior; see, for example, Albert and Chib (1993), Holmes and Held (2006), Frühwirth-Schnatter and Frühwirth (2010), Polson, Scott and Windle (2013).

To prove that such a specialised algorithm is useful, it seems important to show that it may perform better than a generic algorithm, at least in certain cases. The aforementioned papers mostly consider small datasets (essentially those discussed in Section 5.1, which have about 20 predictors or less). We have seen, however, that, for example, importance sampling and random walk Metropolis are hard to beat on such small datasets. Our recommendation, to anyone who is developing a new specialised algorithm, such as a Gibbs sampler for logistic regression, would be to test it on either a much bigger dataset (more than 100 predictors), or on a similar but slightly more complex model [such as a hierarchical logistic regression, as in Section 3.2 of Frühwirth-Schnatter et al., 2009]. We also recommend to systematically compare the proposed algorithm to simple generic methods, such as properly calibrated versions of importance sampling and random walk Metropolis.

7.2.2 Benchmarking generic algorithms. By “generic algorithm,” we mean in particular sampling-based algorithms such as Metropolis–Hastings, HMC and SMC which may sample from any target distribution. For such algorithms, a “benchmark” amounts to the choice of a target distribution.

What our numerical study seems to indicate is that, as far as generic algorithms are concerned, sampling

from the posterior distribution of a binary regression model and a small dataset may not be much more challenging than sampling from a Gaussian of the same dimension.

Of course, it is still useful to showcase a new generic algorithm on such a posterior, if only as a “sanity check” (as pointed out by two reviewers), that is: (a) to quickly check it simulates from the correct distribution; and (b) as a sensible place to start; that is, if the proposed algorithm works well on such a posterior, then it motivates looking at more challenging target distributions.

To construct more challenging posteriors from binary regression models, interesting approaches may be: having more predictors ($p > 100$), having strongly correlated predictors (by, e.g., introducing cross effects), and making the posterior less Gaussian through, for example, a spike-and-slab prior. Alternatively, using a PAC-Bayesian posterior such as, for example, (2.5), the log-density of which is not concave and nondifferentiable, could be an interesting option.

7.3 Big Data and the p^3 Frontier

Several recent papers (Wang and Dunson, 2013, Scott et al., 2016, Bardenet, Doucet and Holmes, 2015) have approached the “big data” problem in Bayesian computation by focussing on the big $n_{\mathcal{D}}$ (many observations) scenario. In binary regression, and possibly in similar models, the big p problem (many covariates) seems more critical, as the complexity of most the algorithms we have discussed is $\mathcal{O}(n_{\mathcal{D}}p^3)$. Indeed, we do not believe that any of the methods discussed in this paper is practical for $p \gg 1000$. The large p problem may be therefore the current frontier of Bayesian computation for binary regression.

Perhaps one way to address the large p problem is to make stronger approximations; for instance, by using EP with an approximation family of sparse Gaussians. Alternatively, one may use a variable selection prior that forbids the number of active covariates to be larger than a certain threshold.

8. GENERALISING TO OTHER MODELS

One should of course remain cautious regarding how our findings may extend to other classes of models. Some of the methods (e.g., Gibbs) which did not perform best for binary regression may turn out in other contexts to be the only viable option.

That said, there are two aspects of our study which we recommend to consider more generally when

studying other models: parallelisation, and taking into account the availability of fast approximations. The former has already been discussed. Regarding the latter, binary regression models are certainly not the only models such that some fast approximations may be obtained, whether through Laplace, INLA, Variational Bayes or EP. And using this approximation to calibrate sampling-based algorithms (Hastings–Metropolis, HMC, SMC and so on) will often have a dramatic impact on the relative performance of these algorithms. One may also discover, at least in certain cases, that these approximations may be sufficiently accurate to be used directly.

ACKNOWLEDGEMENTS

We thank Simon Barthelmé, Manuel Fernández Delgado, Christian P. Robert, Håvard Rue, Ingmar Schuster, the Editor, the Associate Editor and the two anonymous referees for their helpful comments. The first author is partially funded by Labex ECODEC ANR-11-LABEX-0047 grant from ANR (Agence Nationale de la Recherche).

SUPPLEMENTARY MATERIAL

Supplement to “Leave Pima India Alones: Binary regression as a benchmark for Bayesian computation” (DOI: [10.1214/16-STSS581SUPP](https://doi.org/10.1214/16-STSS581SUPP); .pdf). This document contains extra results for paper “Leave Pima Indians alone: binary regression as a benchmark for Bayesian computation”, and provides additional details on the implementation of the EP and SMC algorithms.

REFERENCES

- ALBERT, J. H. and CHIB, S. (1993). Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.* **88** 669–679. [MR1224394](#)
- ALQUIER, P., RIDGWAY, J. and CHOPIN, N. (2016). On the properties of variational approximations of Gibbs posteriors. *J. Mach. Learn. Res.* **17** 1–41.
- ANDRIEU, C. and ROBERTS, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Statist.* **37** 697–725. [MR2502648](#)
- ANDRIEU, C. and THOMS, J. (2008). A tutorial on adaptive MCMC. *Stat. Comput.* **18** 343–373. [MR2461882](#)
- ATTIAS, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* 21–30. Morgan Kaufmann, Stockholm, Sweden.
- BARDENET, R., DOUCET, A. and HOLMES, C. (2015). On Markov chain Monte Carlo methods for tall data. Preprint. Available at [arXiv:1505.02827](https://arxiv.org/abs/1505.02827).

- BESKOS, A., PILLAI, N., ROBERTS, G., SANZ-SERNA, J.-M. and STUART, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli* **19** 1501–1534. [MR3129023](#)
- BISHOP, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York. [MR2247587](#)
- BISSIRI, P., HOLMES, C. and WALKER, S. (2013). A general framework for updating belief distributions. Preprint. Available at [arXiv:1306.6430](#).
- BLEI, D. M., KUCUKELBIR, A. and MCAULIFFE, J. D. (2016). Variational inference: A review for Statisticians. Preprint. Available at [arXiv:1601.00670](#).
- CARPENTER, J., CLIFFORD, P. and FEARNHEAD, P. (1999). Improved particle filter for nonlinear problems. *IEE Proc. Radar, Sonar Navigation* **146** 2–7.
- CATONI, O. (2004). *Statistical Learning Theory and Stochastic Optimization. Lecture Notes in Math.* **1851**. Springer, Berlin. [MR2163920](#)
- CATONI, O. (2007). *Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning. Institute of Mathematical Statistics Lecture Notes—Monograph Series* **56**. IMS, Beachwood, OH. [MR2483528](#)
- CHIPMAN, H., GEORGE, E. I. and MCCULLOCH, R. E. (2001). The practical implementation of Bayesian model selection. In *Model Selection. Institute of Mathematical Statistics Lecture Notes—Monograph Series* **38** 65–134. IMS, Beachwood, OH. [MR2000752](#)
- CHOPIN, N. (2002). A sequential particle filter method for static models. *Biometrika* **89** 539–551. [MR1929161](#)
- CHOPIN, N. (2011). Fast simulation of truncated Gaussian distributions. *Stat. Comput.* **21** 275–288. [MR2774857](#)
- CHOPIN, N., JACOB, P. E. and PAPASPILIOPOULOS, O. (2013). SMC²: An efficient algorithm for sequential analysis of state space models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **75** 397–426. [MR3065473](#)
- CHOPIN, N. and RIDGWAY, J. (2016). Supplement to “Leave Pima Indians Alone: Binary Regression as a Benchmark for Bayesian Computation.” DOI:10.1214/16-ST581SUPP.
- CONSONNI, G. and MARIN, J.-M. (2008). Mean-field variational approximate Bayesian inference for latent variable models. *Comput. Statist. Data Anal.* **52** 790–798. [MR2418528](#)
- COOK, J. D. (2014). Time exchange rate. *The Endeavour*. (Blog).
- DEHAENE, G. P. and BARTHELMÉ, S. (2015a). Bounding errors of expectation-propagation. In *Advances in Neural Information Processing Systems* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, eds.) **28** 244–252. Curran Associates, Red Hook.
- DEHAENE, G. P. and BARTHELMÉ, S. (2015b). Expectation Propagation in the large-data limit. Preprint. Available at [arXiv:1503.08060](#).
- DEL MORAL, P., DOUCET, A. and JASRA, A. (2006). Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **68** 411–436. [MR2278333](#)
- DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **39** 1–38. With discussion. [MR0501537](#)
- DUANE, S., KENNEDY, A., PENDLETON, B. J. and ROWETH, D. (1987). Hybrid Monte Carlo. *Phys. Lett. B* **195** 216–222.
- FAES, C., ORMEROD, J. T. and WAND, M. P. (2011). Variational Bayesian inference for parametric and nonparametric regression with missing data. *J. Amer. Statist. Assoc.* **106** 959–971. [MR2894756](#)
- FIRTH, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80** 27–38. [MR1225212](#)
- FRÜHWIRTH-SCHNATTER, S. and FRÜHWIRTH, R. (2010). Data augmentation and MCMC for binary and multinomial logic models. In *Statistical Modelling and Regression Structures* 111–132. Physica-Verlag/Springer, Heidelberg. [MR2664631](#)
- FRÜHWIRTH-SCHNATTER, S., FRÜHWIRTH, R., HELD, L. and RUE, H. (2009). Improved auxiliary mixture sampling for hierarchical models of non-Gaussian data. *Stat. Comput.* **19** 479–492. [MR2565319](#)
- GELMAN, A. and HILL, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge Univ. Press, Cambridge.
- GELMAN, A., JAKULIN, A., PITTAU, M. G. and SU, Y.-S. (2008). A weakly informative default prior distribution for logistic and other regression models. *Ann. Appl. Stat.* **2** 1360–1383. [MR2655663](#)
- GEORGE, E. I. and MCCULLOCH, R. E. (1993). Variable selection via Gibbs sampling. *J. Amer. Statist. Assoc.* **88** 881–889.
- GHAHRAMANI, Z. and BEAL, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems* (S. A. Solla, T. K. Leen and K. Müller, eds.) **12** 449–455. MIT Press, Cambridge.
- GIROLAMI, M. and CALDERHEAD, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **73** 123–214. [MR2814492](#)
- GRAMACY, R. B. and POLSON, N. G. (2012). Simulation-based regularized logistic regression. *Bayesian Anal.* **7** 567–589. [MR2981628](#)
- GREGOR, K., DANIHELKA, I., GRAVES, A., REZENDE, D. J. and WIERSTRA, D. (2015). Draw: A recurrent neural network for image generation. Preprint. Available at [arXiv:1502.04623](#).
- HERNÁNDEZ-LOBATO, D., HERNÁNDEZ-LOBATO, J. M. and DUPONT, P. (2013). Generalized spike-and-slab priors for Bayesian group feature selection using expectation propagation. *J. Mach. Learn. Res.* **14** 1891–1945. [MR3104499](#)
- HOFFMAN, M. D. and BLEI, D. M. (2014). Structured stochastic variational inference. Preprint. Available at [arXiv:1404.4114](#).
- HOFFMAN, M. D. and GELMAN, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **15** 1593–1623. [MR3214779](#)
- HOFFMAN, M. D., BLEI, D. M., WANG, C. and PAISLEY, J. (2013). Stochastic variational inference. *J. Mach. Learn. Res.* **14** 1303–1347. [MR3081926](#)
- HOLMES, C. C. and HELD, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Anal.* **1** 145–168. [MR2227368](#)
- HÖRMANN, W. and LEYDOLD, J. (2005). Quasi importance sampling. Technical report.
- JACOB, P., ROBERT, C. P. and SMITH, M. H. (2011). Using parallel computation to improve independent Metropolis–Hastings based estimation. *J. Comput. Graph. Statist.* **20** 616–635. [MR2878993](#)
- JASRA, A., STEPHENS, D. A., DOUCET, A. and TSAGARIS, T. (2011). Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo. *Scand. J. Stat.* **38** 1–22. [MR2760137](#)

- KABÁN, A. (2007). On Bayesian classification with Laplace priors. *Pattern Recognition Letters* **28** 1271–1282.
- KHAN, M. E., ARAVKIN, A., FRIEDLANDER, M. and SEEGER, M. (2013). Fast dual variational inference for non-conjugate latent Gaussian models. In *Proceedings of the 30th International Conference on Machine Learning, Number EPFL-CONF-186800*. Omni Press, Atlanta, Georgia.
- KINGMA, D. P. and WELING, M. (2013). Auto-encoding variational bayes. Preprint. Available at [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
- KONG, A., LIU, J. S. and WONG, W. H. (1994). Sequential imputation and Bayesian missing data problems. *J. Amer. Statist. Assoc.* **89** 278–288.
- LAMNISOS, D., GRIFFIN, J. E. and STEEL, M. F. J. (2013). Adaptive Monte Carlo for Bayesian variable selection in regression models. *J. Comput. Graph. Statist.* **22** 729–748. [MR3173739](https://arxiv.org/abs/1312.6114)
- LEE, A., YAU, C., GILES, M. B., DOUCET, A. and HOLMES, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *J. Comput. Graph. Statist.* **19** 769–789.
- LEMIEUX, C. (2009). *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer, New York. [MR2723077](https://arxiv.org/abs/1312.6114)
- MARIN, J.-M. and ROBERT, C. P. (2011). Importance sampling methods for Bayesian discrimination between embedded models. In *Frontiers of Statistical Decision Making and Bayesian Analysis* (M.-H. Chen, D. Dey, P. Muller, D. Sun and K. Ye, eds.). Springer, New York.
- MCALESTER, D. A. (1998). Some PAC-Bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (Madison, WI, 1998)* 230–234 (electronic). ACM, New York. [MR1811587](https://arxiv.org/abs/1312.6114)
- MINKA, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *UAI'01 Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence* **17** 362–369. Morgan Kaufmann, Seattle, WA.
- MINKA, T. (2004). Power EP. Technical report, Dept. Statistics, Carnegie Mellon University, Pittsburgh, PA.
- NEAL, R. M. (2001). Annealed importance sampling. *Stat. Comput.* **11** 125–139. [MR1837132](https://arxiv.org/abs/1312.6114)
- NEAL, R. M. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo. Chapman & Hall/CRC Handb. Mod. Stat. Methods* 113–162. CRC Press, Boca Raton, FL. [MR2858447](https://arxiv.org/abs/1312.6114)
- NICKISCH, H. and RASMUSSEN, C. E. (2008). Approximations for binary Gaussian process classification. *J. Mach. Learn. Res.* **9** 2035–2078. [MR2452620](https://arxiv.org/abs/1312.6114)
- OPPER, M. and ARCHAMBEAU, C. (2009). The variational Gaussian approximation revisited. *Neural Comput.* **21** 786–792. [MR2478318](https://arxiv.org/abs/1312.6114)
- POLSON, N. G., SCOTT, J. G. and WINDLE, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Amer. Statist. Assoc.* **108** 1339–1349. [MR3174712](https://arxiv.org/abs/1312.6114)
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. and FLANNERY, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge Univ. Press, Cambridge. [MR2371990](https://arxiv.org/abs/1312.6114)
- RANGANATH, R., GERRISH, S. and BLEI, D. M. (2014). Black box variational inference. In *AISTATS* 814–822.
- REZENDE, D. J., MOHAMED, S. and WIERSTRA, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *AISTAT*.
- RIDGWAY, J. (2016). Computation of Gaussian orthant probabilities in high dimension. *Stat. Comput.* **26** 899–916. [MR3515028](https://arxiv.org/abs/1312.6114)
- RIDGWAY, J., ALQUIER, P., CHOPIN, N. and LIANG, F. (2014). PAC-Bayesian AUC classification and scoring. In *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger, eds.) **27** 658–666. Curran Associates, Red Hook.
- ROBERT, C. P. and CASELLA, G. (2004). *Monte Carlo Statistical Methods*, 2nd ed. Springer, New York. [MR2080278](https://arxiv.org/abs/1312.6114)
- ROBERTS, G. O. and ROSENTHAL, J. S. (2001). Optimal scaling for various Metropolis–Hastings algorithms. *Statist. Sci.* **16** 351–367. [MR1888450](https://arxiv.org/abs/1312.6114)
- RUE, H., MARTINO, S. and CHOPIN, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **71** 319–392. [MR2649602](https://arxiv.org/abs/1312.6114)
- SCHÄFER, C. (2012). Monte Carlo methods for sampling high-dimensional binary vectors. Ph.D. thesis, Université Paris Dauphine.
- SCHÄFER, C. and CHOPIN, N. (2013). Sequential Monte Carlo on large binary sampling spaces. *Stat. Comput.* **23** 163–184. [MR3016936](https://arxiv.org/abs/1312.6114)
- SCOTT, S. L., BLOCKER, A. W., BONASSI, F. V., CHIPMAN, H., GEORGE, E. and MCCULLOCH, R. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management* **11** 78–88.
- SEEGER, M. (2005). Expectation propagation for exponential families. Technical report, Univ. California Berkeley.
- SEEGER, M., GERWINN, S. and BETHGE, M. (2007). Bayesian inference for sparse generalized linear models. In *European Conference on Machine Learning* 298–309. Springer, Berlin.
- SHAHBABA, B., LAN, S., JOHNSON, W. O. and NEAL, R. M. (2014). Split Hamiltonian Monte Carlo. *Stat. Comput.* **24** 339–349. [MR3192260](https://arxiv.org/abs/1312.6114)
- SHAWE-TAYLOR, J. and WILLIAMSON, R. (1997). A PAC analysis of a Bayesian estimator. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory* 2–9. ACM, New York.
- SKILLING, J. (2006). Nested sampling for general Bayesian computation. *Bayesian Anal.* **1** 833–859 (electronic). [MR2282208](https://arxiv.org/abs/1312.6114)
- SUCHARD, M. A., WANG, Q., CHAN, C., FRELINGER, J., CRON, A. and WEST, M. (2010). Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *J. Comput. Graph. Statist.* **19** 419–438. [MR2758309](https://arxiv.org/abs/1312.6114)
- TIERNEY, L. and KADANE, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *J. Amer. Statist. Assoc.* **81** 82–86. [MR0830567](https://arxiv.org/abs/1312.6114)
- TIERNEY, L., KASS, R. E. and KADANE, J. B. (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *J. Amer. Statist. Assoc.* **84** 710–716. [MR1132586](https://arxiv.org/abs/1312.6114)
- VAN GERVEN, M. A. J., CSEKE, B., DE LANGE, F. P. and HESKES, T. (2010). Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *Neuroimage* **50** 150–161.
- WAINWRIGHT, M. J. and JORDAN, M. I. (2008). Graphical models, exponential families, and variational inference. *Faund. Trends Mach. Learn.* **1** 1–305.
- WANG, X. and DUNSON, D. B. (2013). Parallelizing MCMC via Weierstrass sampler. Preprint. Available at [arXiv:1312.4605](https://arxiv.org/abs/1312.6114).