On the Fully Bayesian Treatment of Latent Gaussian Models using Stochastic Simulations

Maurizio Filippone[1], Mingjun Zhong[2], Mark Girolami[3]

1 – School of Computing Science, University of Glasgow
2 – Department of Biomedical Engineering, Dalian University of Technology,
3 – Department of Statistical Science, University College London,
email corresponding author - maurizio.filippone@glasgow.ac.uk

February 2012

School of Computing Science - University of Glasgow
Technical Report TR-2012-329

Abstract

Latent Gaussian models (LGMs) are extensively used in data analysis given their flexible modeling capabilities and interpretability. The fully Bayesian treatment of LGMs is usually intractable, and therefore it is necessary to resort to approximations. This paper proposes the use of stochastic simulations based on Markov chain Monte Carlo (MCMC) methods for small to moderately sized data sets and for LGMs comprising a set of parameters that prevents the use of quadrature techniques. We discuss the challenges in applying MCMC methods to LGMs and compare different strategies based on efficient parametrizations and efficient proposal mechanisms. Extensive evaluation on simulated and real data suggests a sampling strategy that achieves high efficiency with moderate cost compared to state-of-the-art methods for the fully Bayesian treatment of LGMs.

1 Introduction

Bayesian inference represents one of the important paradigms for data analysis as it provides powerful tools to carry out inference in complex models. In a Bayesian treatment, all quantities in the model are interpreted as random variables, and probability distributions over parameters of interest are obtained after data are observed. In contrast with point estimates, this allows to quantify uncertainty in parameter estimates and to obtain predictive distributions thus making it possible to balance the cost of decisions. Although the probabilistic framework yields a rich and highly interpretable model description, from the computational perspective the inference problem in complex systems is usually challenging. In particular, the inference problem amounts in solving integrals of functions that, in typical real applications, are defined on large dimensional spaces and are analytically intractable. In order to get around this intractability, deterministic approximations or approximations based on stochastic simulations have been proposed in the literature (see, e.g., Bishop (2007) for an overview).

In this paper, we will focus on Latent Gaussian Models (LGMs), which is a class of models that is fairly popular in data analysis due to the associated interpretability and flexibility. From the modeling perspective, LGMs are quite simple and akin to Generalized Linear Models (Nelder and Wedderburn, 1972). In LGMs, observations (e.g., class labels or counts) are associated to input data (e.g., patients or spatial locations) described by means of a set of covariates. Observations are assumed conditionally independent given a set of latent variables, and distributed according to a certain distribution depending on the particular type of data, e.g., Bernoulli for binary labels and Poisson for observations in the form of counts. Latent variables are assigned a Gaussian Process (GP) prior, which implies joint normality due to the properties of marginals of Gaussian distributions. The covariance structure of the latent variables is then parametrized by a set of (hyper)-parameters that characterizes the covariance of the input data in terms of length-scales and intensity of interaction. In fact, the prior over the latent variables can be interpreted as a prior over functions, and hyper-parameters specify their characteristics, such as degree of smoothness and range of spanned values. The model structure is therefore hierarchical, with hyper-parameters conditioning the latent variables that, in turn, condition observations. LGMs comprise a large set of models, and in this paper we will focus in particular on Logistic Regression with GP priors, Gaussian Copula Process Volatility model, Log-Gaussian Cox model, and Ordinal Regression with GP priors.

Exact inference in LGMs is analytically intractable, as the likelihood of the observations is usually not conjugate to the Gaussian prior over the latent functions, and hyper-parameters cannot be analytically integrated out either. Deterministic approximations, such as the Laplace Approximation (LA) (Tierney and Kadane, 1986) or Expectation Propagation (EP) (Minka, 2001) have been proposed for integrating out latent functions in LGMs, and in particular for classification using GP priors (see Kuss and Rasmussen (2005) for an extensive assessment of these approximations for binary classification with GP priors). These approximations provide a computationally tractable way to integrate out latent functions, but unfortunately it is not possible to quantify the error introduced by these approximations. LA in particular can yield a very rough approximation of the target density and can lead to a poor estimation of predictive means and variances (Kuss and Rasmussen, 2005). In this respect, approximations based on moment matching, such as EP, seem to be superior, but from the theoretical perspective there is no guarantee of convergence for EP. Recently, new advances in the field of approximate inference for LGMs were proposed in Rue et al. (2009) with the use of Integrated Nested Laplace Approximation (INLA), and in Cseke and Heskes (2011) that proposes refinements to EP. Such advances provide a step forward in approximate methods, but there are still a number of limitations, as these methods target the integration of latent functions only.

In the direction of providing a fully Bayesian treatment of LGMs, it is necessary to integrate out the hyper-parameters as well, and this, combined with the integration of the latent variables is a challenging task. INLA, for example, relies upon a numerical quadrature based on a gridding scheme that can therefore be applied to problems with a relatively low set parameters (less than six as stated in Rue et al. (2009)). Therefore, in cases where the number of hyper-parameters is large, INLA cannot be directly applied. The same considerations apply to EP, that allows to integrate out the latent variables, but needs to be embedded into a scheme to integrate out the hyper-parameters, and again numerical quadrature are usually adopted (Cseke and Heskes, 2011). Finally, the application of approximations using Markov Chain Monte Carlo (MCMC) methods is complicated by the hierarchical structure of LGMs that makes parameters and latent functions strongly coupled a posteriori, thus making chains converge slowly and mix poorly (Murray and Adams, 2010).

The aim of this paper is to provide a simple and widely applicable methodology that can be applied to carry out the fully Bayesian treatment of LGMs. We propose to tackle intractability in Bayesian inference in LGMs through the use of stochastic simulations based on Markov Chain Monte Carlo (MCMC) methods to sample from the posterior distribution of parameters and latent variables. This allows to solve the integration needed to obtain the predictive distribution for new samples by means of Monte Carlo integration (Robert and Casella, 2005). The use of MCMC methods is motivated by the fact that they are quite flexible and allow to exploit asymptotic guarantees of convergence of associated Monte Carlo estimates. However, the rate of convergence to zero of the variance of Monte Carlo estimates is in the order of the inverse of the number of independent samples used to compute them, and this can require large number of independent samples in some applications. In the case of MCMC methods, samples will have a certain degree of correlation given by their efficiency and ability to explore the whole space of parameters. This motivates the need for MCMC strategies that allow to reduce correlation within the chains and that converge quickly to the target distribution. This is quite a challenging task, especially in the case of LGMs. In most cases, sampling involves some sort of proposal mechanism and an accept/reject rule. Examples of such MCMC methods comprise the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) which is based on a random walk type of proposal mechanism. In this case, it is required to specify the covariance of the proposal distribution which drives the random walk exploration of the space, and the proposals are accepted and rejected so as to let the chain sample from the correct invariant distribution. Random walk exploration tends to be quite inefficient, and methods based on diffusion (Langevin Adjusted Metropolis Algorithm (MALA) (Roberts and Rosenthal, 1998)), or methods based on Hamiltonian mechanics (Hybrid Monte Carlo (HMC) (Neal, 1993)) offer an improvement given by the fact that gradients drive the exploration toward regions of higher density. In these methods it is necessary to specify a so called *mass matrix*, whose size is quadratic in the number of parameters. Given the the aim is to draw independent samples from the target distribution, it is very important to tune parameters of proposal distributions in MCMC methods in order to avoid strong correlations within the chain, or the possibility that the chain does not move at all. In several dimensions, it is clear that the design of effective proposal mechanism becomes challenging as several parameters need to be tuned. Manifold methods (Girolami and Calderhead, 2011) provide a systematic way of designing proposals for MALA and HMC by making use of the natural geometry of the statistical manifold; one of the goals of this paper is to assess the effectiveness of these proposal mechanisms for inference in LGMs.

To summarize, the application of MCMC methods for efficiently sampling from the posterior of latent functions and hyper-parameters in LGMs is rather challenging, and the reasons are mainly two: (i) the model structure makes hyper-parameters and latent functions strongly coupled a pos-

teriori (Murray and Adams, 2010; Neal, 1999) and (ii) very high dimensional integrals have to be estimated.

(i) Due to the hierarchical structure of LGMs, chains converge slowly and mix poorly if the coupling effect between the groups of variables is not dealt with properly. This effect has drawn a lot of attention in the case of hierarchical models in general (Papaspiliopoulos et al., 2007; Yu and Meng, 2011), and recently in latent Gaussian models. The main contributions that propose ways of tackling this problem in LGMs are presented in Knorr-Held and Rue (2002); Murray and Adams (2010). In Knorr-Held and Rue (2002) a joint update of latent variables and hyper-parameters is proposed with the aim of avoiding proposals for hyper-parameters to be conditioned on the values of latent variables. In Murray and Adams (2010) an improved parametrization is proposed and coupled with the update of hyper-parameters using a component-wise slice sampling strategy. In both cases, the sampling efficiency compared to the complexity is still an issue and we are interested in studying whether it is possible to improve upon these approaches.

(ii) Sampling hyper-parameters and latent functions cannot be done using exact Gibbs steps, and it requires proposals that are accepted/rejected based on a Metropolis ratio, leading to a waste of expensive computations. Samplers characterized by acceptance mechanisms embedded in a Gibbs sampler, are usually referred to Metropolis-within-Gibbs samplers. Designing proposals that guarantee high acceptance and independence between samples becomes extremely difficult, especially because in LGMs latent functions can have dimensions in the order of hundreds or thousands.

In response to (i), we will investigate the use of reparametrization techniques, that have been studied in hierarchical models with the aim of achieving faster convergence (see, e.g., Papaspiliopoulos et al. (2007)). In the terminology of Yu and Meng (2011), we identify two particular cases, namely Sufficient Augmentation (SA), and Ancillary Augmentation (AA). In SA, the latent variables are a sufficient statistic for the observed data, whereas in the AA latent variables are transformed via the hyper-parameters, so that they are ancillary. In this work, we make use of reparametrization techniques, and in particular a combination of reparametrization techniques based on interweaving SA and AA as illustrated in Yu and Meng (2011), to overcome the problem of coupling between latent functions and hyper-parameters. We will use the recently presented Ancillarity-Sufficiency Interweaving Strategy (ASIS) to break the correlation between hyper-parameters and latent variables. ASIS combines two complementary reparametrizations to break such a correlation. SA and AA and are complementary in the sense that have better performances in either strong or weak data limits. The idea behind ASIS is to have a method that can sample efficiently in both strong and weak data scenarios. In essence the proposed sampling scheme is rather simple, as it is a Gibbs sampler that iterates between updates of the latent functions and interweaving two updating schemes for the hyper-parameters. The aim of this paper is to test whether combining different parametrizations allows to achieve efficient sampling in LGMs. In the experiments, we include the state-of-the-art methods presented in Knorr-Held and Rue (2002); Murray and Adams (2010) in the comparisons to have baseline performance against which we can assess AA, SA, and ASIS.

In response to (ii), we will compare different Metropolis-within-Gibbs samplers based on different principles with the aim of understanding which samplers offer the best performance, in terms of efficiency relative to the computational cost, for different steps of the Gibbs sampler. Among the samplers that we will consider, we will compare methods that make use of the gradient of the target distribution and methods that make use of the curvature as given by the Fisher Information (manifold methods). Given the computational complexity of some of the considered methods we also study a few variations that lower the computational complexity while retaining high efficiency.

By combining efficient Metropolis-within-Gibbs samplers and reparametrization techniques, we

will show that it is possible to improve efficiency with respect to state-of-the-art sampling schemes, making a step forward in the applicability of MCMC techniques for the fully Bayesian treatment of LGMs[1]. The paper is organized as follows: Section 2 introduces LGMs, section 3 reports the ideas underpinning the interweaving strategy in Yu and Meng (2011), while section 4 describes the samplers employed in this work. Sections 5 and 6 report extensive results and comparisons of different sampling strategies on simulated and real data respectively, and section 7 concludes the paper. For the sake of readability, most of the technical derivations and tables of results can be found in the appendices.

2 Latent Gaussian Models - (LGMs)

In this section, we give a general formulation of LGMs. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n input data described by a set of d covariates $\mathbf{x}_i \in \mathbb{R}^d$, associated with observed responses $\mathbf{y} = \{y_1, \ldots, y_n\}$. Let k be the covariance function modeling the covariance structure between latent variables, parametrized by a vector of (hyper)-parameters $\boldsymbol{\theta} = (\sigma, \psi_{\tau_1}, \ldots, \psi_{\tau_d})$:

$$k(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}) = \sigma q(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\psi}_{\boldsymbol{\tau}}) = \sigma \left(\exp \left[-\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} A (\mathbf{x}_i - \mathbf{x}_j) \right] + \omega \delta_{ij} \right)$$

with the matrix A defining the length-scale of the interaction between the input data. In particular, the covariance can be diagonal to allow for different scaling of the covariates (also used for Automatic Relevance Determination (ARD) (Mackay, 1994; Neal, 1996) of the covariates) $A^{-1} = \mathrm{diag}\left(\exp(\psi_{\tau_1}), \ldots, \exp(\psi_{\tau_d})\right)$ or simply isotropic $A^{-1} = \exp(\psi_\tau) I$. Note that in both cases the values $\tau_i = \sqrt{\exp(\psi_{\tau_i})}$ can be interpreted as length-scale parameters. This definition of covariance function is adopted in many applications and is the one we will consider in the remainder of this paper. Note also that length-scales have to be positive and exponentiation will be convenient when sampling using MCMC methods. In general, sampling positives variables using MH can be done by using Gamma types of proposals (Robert and Casella, 2005), and in Hamiltonian based samplers by using boundary conditions (Neal, 2010), but we will not consider these methods here. We prefer instead to keep the standard unconstrained proposal mechanisms of the considered samplers by working in log-space.

Let Q be the matrix whose entries are $q_{ij} = q(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\psi}_{\boldsymbol{\tau}})$; the covariance matrix K will then be:

$$K = \sigma Q$$

The parameter ω is a small jitter that improve the conditioning of Q.

Consider the following general form of LGMs to model the generative process of the observed $\mathbf{y}$ given X (note that we are not including any parameters for a mean function of the GP and extra parameters for the transformation $\zeta()$):

$p(\boldsymbol{\theta})$	prior over the hyper-parameters
$Q = LL^T$	decomposition correlation matrix
$p(\boldsymbol{\nu}) \sim \mathcal{N}(0, I)$	whitened latent variables
$\mathbf{f} = \sqrt{\sigma} L \boldsymbol{\nu}$	transformation to obtain the latent function
$\Downarrow$	
$p(\mathbf{f}\|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}\|\mathbf{0}, K)$	latent function
$p(\mathbf{y}\|\mathbf{f}) = \prod_{i=1}^{n} \mathcal{E}(y_i\|\zeta(f_i))$	likelihood observations

[1] An implementation of the methods considered in this paper can be found at:
http://www.dcs.gla.ac.uk/~maurizio/pages/code.html

Note that for the sake of convenience, in this work we make use of the Cholesky factorization algorithm (Golub and Van Loan, 1996) whenever we need to compute square roots of matrices, so that L will be lower triangular. For generality, the distribution of the observed random variables $\mathbf{y}$ is expressed in terms of the exponential family of distributions $\mathcal{E}$ with natural parameters given by a transformed version of the latent variables via $\zeta()$. The notation that we will use to denote the form of the exponential family of distributions is:

$$\mathcal{E}(y|\zeta(f)) = h(y)g[\zeta(f)]\exp[\zeta(f)u(y)]$$

The exponential family of distributions comprises many of the commonly used distributions, such as the Bernoulli, the Poisson, the Gaussian, and the exponential, just to name a few. Such a general likelihood form will allow to express in a general way the key quantities needed to use the samplers that we will employ in this work.

In a Bayesian setting, the predictive distribution for a new test sample $\mathbf{x}_*$ can be written in the following way (for the sake of clarity we drop the explicit conditioning on X and $\mathbf{x}_*$):

$$p(y_*|\mathbf{y}) = \int\int\int p(y_*|f_*)p(f_*|\mathbf{f},\boldsymbol{\theta})p(\mathbf{f},\boldsymbol{\theta}|\mathbf{y})df_*d\mathbf{f}d\boldsymbol{\theta}$$

This expression is quite powerful for the following reasons. First, it is a probability distribution over the prediction for y_* rather than a point estimate, and it allows to quantify the confidence in predicting y_*. Second, the predictive distribution doesn't contain parameters anymore, as they are integrated out; the mechanism by which they are integrated out is such that the predictive distribution accounts for the uncertainty in parameters estimates after data are observed. Such a source of uncertainty is captured by the posterior distribution $p(\mathbf{f},\boldsymbol{\theta}|\mathbf{y})$, that becomes the key element of inference.

The solution of the integral needed to compute the predictive distribution requires the integration over the latent process and the parameters which is generally analytically intractable. Numerical integration methods are therefore needed, either based on deterministic approximations or stochastic simulations. In this work we will focus on stochastic approximations for obtaining samples from the posterior distribution of $\mathbf{f}$ and $\boldsymbol{\theta}$, so that we can estimate the predictive distribution using the Monte Carlo method:

$$p(y_*|\mathbf{y}) \simeq \frac{1}{N}\sum_{i=1}^{N}\int p(y_*|f_*)p(f_*|\mathbf{f}^{(i)},\boldsymbol{\theta}^{(i)})df_*$$

In this expression we denoted by $\mathbf{f}^{(i)}$ and $\boldsymbol{\theta}^{(i)}$ the i-th independent samples from the posterior distribution of $\mathbf{f}$ and $\boldsymbol{\theta}$ that we will obtain by means of Markov chain Monte Carlo (MCMC) methods. Note that MCMC methods obtain posterior samples that have a certain degree of correlation and therefore we will need to sub-sample (thin) the Markov chains according to their auto-correlation. In particular, we will evaluate the Effective Sample Size (ESS) (Gilks and Spiegelhalter, 1996; Robert and Casella, 2005) and assess different sampling strategy based on this score.

Finally, note that the remaining integral is univariate and it is generally easy to evaluate, as $p(f_*|\mathbf{f}^{(i)},\boldsymbol{\theta}^{(i)})$ is Gaussian $\mathcal{N}(f_*|m_*^{(i)},v_*^{(i)})$ with mean and variance given by:

$$m_*^{(i)} = \mathbf{k}_*^{\mathrm{T}}K^{-1}\mathbf{f}^{(i)} \qquad v_*^{(i)} = k_{**} - \mathbf{k}_*^{\mathrm{T}}K^{-1}\mathbf{k}_*$$

with $k_{**} = k(\mathbf{x}_*,\mathbf{x}_*|\boldsymbol{\theta}^{(i)})$ and $(\mathbf{k}_*)_h = k(\mathbf{x}_*,\mathbf{x}_h|\boldsymbol{\theta}^{(i)})$. Again, using a Monte Carlo estimator:

$$\int p(y_*|f_*)p(f_*|\mathbf{f}^{(i)},\boldsymbol{\theta}^{(i)}) \simeq \frac{1}{N_2}\sum_{j=1}^{N_2}p(y_*|f_*^{(j)})$$

with $f_*^{(j)} \sim \mathcal{N}(f_*^{(j)}|m_*^{(i)}, v_*^{(i)})$.

3 Ancillarity-Sufficiency Interweaving Strategy - ASIS

In this section we briefly review the main results presented in Yu and Meng (2011) on the combination of parametrizations to improve convergence and efficiency of MCMC methods, and we will illustrate how these results can be applied to LGMs. The work by Yu and Meng (2011) is motivated by data augmentation techniques (van Dyk and Meng, 2001), where a set of unobserved (latent) variables is introduced in addition to the parameters of the model. Data augmentation was proposed in problems where the goal is to optimize the likelihood of the observation given the model parameters and the introduction of a set of latent (unobserved) variables makes the optimization easier. Such optimization strategy is the so called Expectation Maximization (Dempster et al., 1977) algorithm, which is an iterative scheme that alternates between the update of parameters and latent functions.

The idea of data augmentation can be been extended to MCMC, but the particular structure of the model makes this task generally difficult. In particular, Gibbs style proposals alternating between updates of latent variables and parameters in the model can be extremely inefficient given the strong coupling a posteriori between the two groups of variables. For this reason, it is important to choose an efficient parametrizations for the particular problem under study and for the available amount of data, as both these aspects can dramatically influence the efficiency and the rate of convergence of the sampler. The work presented by Yu and Meng (2011), provides a study on combinations of parametrizations to cope with this effect. In Yu and Meng (2011) two parametrizations are given particular attention, namely Sufficient Augmentation (SA) and Ancillary Augmentation (AA). In SA, the latent variables are sufficient statistics for the parameters of interest, whereas in AA latent variables are ancillary.

In LGMs, we can identify SA and AA with the parametrization introduced in the former section, and also considered in (Murray and Adams, 2010) where they are referred to unwhitened and whitened respectively. In SA, $\mathbf{f}$ is sufficient for $\mathbf{y}$; for AA, we notice that the generative process offers $\boldsymbol{\nu}$ as an ideal candidate for a set of latent variables that is ancillary for $\mathbf{y}$. The two parametrizations that we will consider, SA and AA, will basically focus on the following log-joint densities:

$$\text{SA} \qquad p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta})$$

$$\text{AA} \qquad p(\mathbf{y}, \boldsymbol{\nu}, \boldsymbol{\theta})$$

Intuitively, combining parametrizations seems to be promising in taking the best of both parametrization, or at least, to avoid the possibility that the chain doesn't converge because of the wrong choice of parametrization. Alternating seems the most obvious way of combining SA and AA, but as recently investigated in Yu and Meng (2011), interweaving SA and AA is actually a good way of combining the two schemes. From the theoretical perspective, the geometric rate of convergence r of the scheme where the parametrizations are interweaved, is related to the rates of the two schemes r_1 and r_2 by

$$r \leq R_{1,2}\sqrt{r_1 r_2}$$

where $R_{1,2}$ is the maximal correlation between the latent variables for the two schemes. Given that the former expression implies $r \leq \max(r_1, r_2)$, we see that combining two parametrizations leads to a scheme that is better than the worst. This is already an advantage compared to using a single scheme when one is in doubt on which scheme to use. However, the key result is the fact that $R_{1,2}$ can be very

small depending on the particular parametrizations chosen, so it is possible to make the combined scheme converge quickly even if none of the two does. In general, this result is quite remarkable, as once different reparametrizations are available, combining them using the interweaving strategy is simple to implement, and can boost sampling efficiency.

In LGMs, the ASIS scheme amounts in interweaving SA and AA updates, that following Yu and Meng (2011) amounts to sample:

$$\mathbf{f}|\mathbf{y},\boldsymbol{\theta} \quad \longrightarrow \quad \boldsymbol{\nu}|\mathbf{f} \quad \longrightarrow \quad \boldsymbol{\theta}|\mathbf{y},\boldsymbol{\nu}$$

Expanding the second step, the Metropolis-within-Gibbs steps follow in this order:

$$\mathbf{f}|\mathbf{y},\boldsymbol{\theta} \quad \longrightarrow \quad \boldsymbol{\theta}|\mathbf{f} \quad \longrightarrow \quad \boldsymbol{\nu} = \sigma^{-1/2}L^{-1}\mathbf{f} \quad \longrightarrow \quad \boldsymbol{\theta}|\mathbf{y},\boldsymbol{\nu}$$

In the next section we present the samplers that we compared in this study to obtain samples from each of these Gibbs steps.

4 Samplers considered in this work

In this section we present the samplers considered in this work. As we noticed in the former sections, it is not possible to derive Gibbs samplers with exact steps, and we have to adopt Metropolis-within-Gibbs type of proposals. We are interested in studying the benefit in terms of efficiency of including gradient or curvature information in the proposal mechanism. We therefore consider samplers with increasing complexity, and in particular Metropolis-Hastings (MH) based on random walk types of proposals, Hybrid Monte Carlo (HMC) which uses gradient information, and manifold methods (Girolami and Calderhead, 2011) which use curvature information.

In order to keep the presentation of the samplers simple, we will focus on the samplers for $\mathbf{f}$, but the same samplers can be easily applied to $\boldsymbol{\theta}$ in both SA and AA. Only Elliptical Slice Sampling has been proposed specifically to sample $\mathbf{f}$ in LGMs, and it doesn't have a counterpart for $\boldsymbol{\theta}$. In the case of sampling the latent functions, the presented samplers aim to sample from the posterior $p(\mathbf{f}|\mathbf{y},\boldsymbol{\theta})$, which is proportional to $p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\theta})$. In the remainder of this section, we will denote $\log[p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\theta})]$ by $W(\mathbf{f})$. Sampling from the posterior of $\boldsymbol{\theta}$ requires the specification of the invariant distribution which is proportional to $p(\mathbf{f}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ in the SA case, and $p(\mathbf{y}|\boldsymbol{\nu},\boldsymbol{\theta})p(\boldsymbol{\theta})$ in the AA case. We can then define $W_{\text{SA}}(\boldsymbol{\theta}) = \log[p(\mathbf{f}|\boldsymbol{\theta})p(\boldsymbol{\theta})]$ and $W_{\text{AA}}(\boldsymbol{\theta}) = \log[p(\mathbf{y}|\boldsymbol{\nu},\boldsymbol{\theta})p(\boldsymbol{\theta})]$ respectively, and apply the samplers presented here for sampling $\boldsymbol{\theta}$ rather than $\mathbf{f}$.

4.1 Metropolis-Hastings - MH

The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) employs a proposal mechanism $q(\mathbf{f}'|\mathbf{f})$ based on a random walk. Tuning the Metropolis-Hastings algorithm involves selecting the right proposal mechanism. A common choice is to use a random walk multivariate Gaussian proposal centered at the former position $\mathbf{f}$ and with covariance Σ, thus taking the form $q(\mathbf{f}'|\mathbf{f}) = \mathcal{N}(\mathbf{f}'|\mathbf{f},\Sigma)$. Proposed moves are accepted with probability

$$\min\left\{1, \frac{\exp(W(\mathbf{f}'))q(\mathbf{f}|\mathbf{f}')}{\exp(W(\mathbf{f}))q(\mathbf{f}'|\mathbf{f})}\right\}$$

that simplifies to $\min\{1, \exp(W(\mathbf{f}') - W(\mathbf{f}))\}$ in the case of symmetric proposals. A sketch of the algorithm is reported in Algorithm 1.

Selecting the covariance matrix however, is far from trivial in most cases since knowledge about the target density is required. Therefore simpler forms of covariance matrices can be used instead, such as isotropic $\Sigma = \epsilon I$. Small values of ϵ imply small transitions and result in high acceptance rates while the mixing of the Markov Chain is poor. Large values on the other hand, allow for large transitions but they result in most of the samples being rejected. Studies on how to optimally tune the MH algorithm are reported e.g., in Roberts and Rosenthal (2001). Adaptive schemes for the MH algorithm have also been proposed (Haario et al., 2005) though they should be applied with care (Andrieu and Thoms, 2008). Note that when sampling $\mathbf{f}$, the matrix Σ is $n \times n$, and its Cholesky

Algorithm 1 MH proposal with Gaussian proposal - $\Sigma = LL^{\mathrm{T}}$

1: $\mathbf{f}' \sim \mathcal{N}(\mathbf{f}'|\mathbf{f}, \Sigma)$ $\{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$ $\mathbf{f}' = L_\Sigma \mathbf{z} + \mathbf{f}\}$
2: $A = \min\{0, W(\mathbf{f}') - W(\mathbf{f})\}$
3: $u \sim \mathcal{E}(u|1)$
4: **if** $r > -u$ **then**
5: **return** $\mathbf{f}'$
6: **else**
7: **return** $\mathbf{f}$
8: **end if**

factorization has complexity in $O(n^3)$ unless simple structures are assumed. This factorization is required only once at the beginning of the algorithm if Σ doesn't change throughout the sampling. Proposing $\mathbf{f}$ with MH, once Σ is factorized, has complexity in $O(n^2)$.

4.2 HMC

In Hybrid Monte Carlo (Duane et al., 1987; Neal, 1993) (HMC) the proposals are based on the analogy with a physical system, where a particle is simulated moving in a potential field. An auxiliary variable $\mathbf{p}$, that plays the role of the momentum variable, is drawn from $\mathcal{N}(\mathbf{p}|\mathbf{0}, M)$. The covariance matrix M is the so called mass matrix. The joint density of $\mathbf{f}$ and $\mathbf{p}$ factorizes as $p(\mathbf{f}, \mathbf{p}) = \exp(W(\mathbf{f}))p(\mathbf{p})$. The negative log-joint density is:

$$H(\mathbf{f}, \mathbf{p}) = -W(\mathbf{f}) + \frac{1}{2}\log(|M|) + \frac{1}{2}\mathbf{p}^{\mathrm{T}} M^{-1}\mathbf{p} + \text{const.}$$

This is the Hamiltonian of the simulated particle, where the potential field is given by $-W(\mathbf{f})$ and the kinetic energy by the quadratic form in $\mathbf{p}$. In order to draw proposals from $p(\mathbf{f}|\mathbf{y}, \boldsymbol{\theta})$, we can simulate the particle for a certain time interval, introducing an analogous of time t and solving Hamilton's equations

$$\frac{d\mathbf{f}}{dt} = \frac{\partial H}{\partial \mathbf{p}} = M^{-1}\mathbf{p}$$

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{f}} = \nabla_{\mathbf{f}} W$$

Given that there are no frictions in the system, the energy will be conserved during the motion of the particle. Solving directly Hamilton's equations for general potential fields, however, is analytically intractable, and therefore it is necessary to resort to schemes where time is discretized in units. The

Leapfrog integrator is one of these schemes that is volume preserving and reversible, and leads to the following discretization:

$$\mathbf{p}_{(t+1/2)} = \mathbf{p}_{(t)} + \frac{\varepsilon}{2}\nabla_{\mathbf{f}} W(\mathbf{f}_{(t)})$$

$$\mathbf{f}_{(t+1)} = \mathbf{f}_{(t)} + \varepsilon M^{-1}\mathbf{p}_{(t+1/2)}$$

$$\mathbf{p}_{(t+1)} = \mathbf{p}_{(1/2)} + \frac{\varepsilon}{2}\nabla_{\mathbf{f}} W(\mathbf{f}_{(t+1)})$$

The number of leapfrog steps L can be randomized (Neal, 1993) and gives an update of $(\mathbf{f}, \mathbf{p})$ into $(\mathbf{f}_{(L)}, \mathbf{p}_{(L)})$. The discretization introduces an approximation such that the total energy is not conserved. In order to ensure that HMC samples from the correct invariant distribution it is needed to adopt a Metropolis accept/reject step:

$$\min\{1, \exp[-H(\mathbf{f}_{(L)}, \mathbf{p}_{(L)}) + H(\mathbf{f}, \mathbf{p})]\}$$

A sketch of the HMC algorithm is reported in Algorithm 2.

Algorithm 2 HMC proposal when $M = L_M L_M^{\mathrm{T}}$

$\mathbf{f}_{(0)} = \mathbf{f}$
$\mathbf{p}_{(0)} \sim \mathcal{N}(\mathbf{p}_{(0)}|\mathbf{0}, M)$ $\{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$ $\mathbf{p}_{(0)} = L_M\mathbf{z}\}$
$L = \text{sample}[1, \ldots, L_{\max}]$
for $t = 0$ to $L - 1$ **do**
    $\mathbf{p}_{(t+1/2)} = \mathbf{p}_{(t)} + \frac{\varepsilon}{2}\nabla_{\mathbf{f}} W(\mathbf{f}_{(t)})$
    $\mathbf{f}_{(t+1)} = \mathbf{f}_{(t)} + \varepsilon M^{-1}\mathbf{p}_{(t+1/2)}$
    $\{M^{-1}\mathbf{p}_{(t+1/2)} = \texttt{backsolve}(L_M^{\mathrm{T}}, (\texttt{forwardsolve}(L_M, \mathbf{p}_{(t+1/2)})))\}$
    $\mathbf{p}_{(t+1)} = \mathbf{p}_{(1/2)} + \frac{\varepsilon}{2}\nabla_{\mathbf{f}} W(\mathbf{f}_{(t+1)})$
end for
$r = \min\left\{0, -H(\mathbf{f}_{(L)}, \mathbf{p}_{(L)}) + H(\mathbf{f}_{(0)}, \mathbf{p}_{(0)})\right\}$
$\{\log|M| = 2\sum_i \log(L_M)_{ii}$ $\quad \mathbf{p}^{\mathrm{T}}M^{-1}\mathbf{p} = \|\texttt{forwardsolve}(L_M, \mathbf{p}_{(L)})\|^2\}$
$u \sim \mathcal{E}(u|1)$
if $r > -u$ **then**
    return $\mathbf{f}_{(L)}$
else
    return $\mathbf{f}_{(0)}$
end if

Similarly to MH, once the Cholesky decomposition of M is available (complexity in $O(n^3)$), drawing posterior samples for $\mathbf{f}$ has complexity in $O(n^2)$. Adaptive schemes for HMC, propose to set M to be an empirical estimate of the inverse of the covariance of posterior samples (Neal, 1993, 1996). Given the large dimensionality of the space where $\mathbf{f}$ lives, this can be problematic in practice, as several samples would be needed before obtaining a reliable estimate.

Note also that combining the update equations of $\mathbf{f}$ and $\mathbf{p}$ for one leapfrog step we obtain a discretized Langevin diffusion as presented in Roberts and Stramer (2002) and termed Metropolis Adjusted Langevin Algorithm (MALA) (see Girolami and Calderhead (2011); Neal (1993) for further discussions on this point).

4.3 HMC specifying the Cholesky of the inverse of the mass matrix

In this section we introduce a modified version of HMC that, rather than using the Cholesky decomposition of the mass matrix, requires the decomposition of its inverse. The reason for introducing such a variation, is that we might want to break the correlation structure between the latent variables by transforming $\mathbf{f}$ using the covariance K or, by employing a geometric argument, using the Fisher Information, which is a function of K (see the appendix). In particular, a sensible choice would be to set $M = K^{-1}$ which is the prior covariance; this would break the correlation between the latent variables a priori. In this case, $M^{-1} = K = \sigma LL^{\mathrm{T}}$; given that L is needed anyway to compute the log-joint density, a definition of HMC requiring the Cholesky factor of the inverse of the mass matrix would be useful. We report this version of HMC in Algorithm 2. Note that once the Cholesky factor of M^{-1} is obtained, drawing posterior samples for $\mathbf{f}$ has complexity in $O(n^2)$.

Algorithm 3 HMC proposal when $M^{-1} = L_{M^{-1}} L_{M^{-1}}^{\mathrm{T}}$

$\mathbf{f}_{(0)} = \mathbf{f}$
$\mathbf{p}_{(0)} \sim \mathcal{N}(\mathbf{p}_{(0)}|\mathbf{0}, M)$ $\{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$ $\mathbf{p}_{(0)} = \texttt{backsolve}(L_{M^{-1}}^{\mathrm{T}}, \mathbf{z})\}$
$L = \text{sample}[1, \ldots, L_{\max}]$
for $t = 0$ to $L - 1$ **do**
    $\mathbf{p}_{(t+1/2)} = \mathbf{p}_{(t)} + \frac{\varepsilon}{2} \nabla_{\mathbf{f}} W(\mathbf{f}_{(t)})$
    $\mathbf{f}_{(t+1)} = \mathbf{f}_{(t)} + \varepsilon M^{-1} \mathbf{p}_{(t+1/2)}$ $\{M^{-1}\mathbf{p}_{(t+1/2)} = L_{M^{-1}}(L_{M^{-1}}^{\mathrm{T}} \mathbf{p}_{(t+1/2)})\}$
    $\mathbf{p}_{(t+1)} = \mathbf{p}_{(1/2)} + \frac{\varepsilon}{2} \nabla_{\mathbf{f}} W(\mathbf{f}_{(t+1)})$
end for
$r = \min\left\{0, -H(\mathbf{f}_{(L)}, \mathbf{p}_{(L)}) + H(\mathbf{f}_{(0)}, \mathbf{p}_{(0)})\right\}$
$\{\log|M| = -2\sum_i \log(L_{M^{-1}})_{ii}$ $\quad \mathbf{p}^{\mathrm{T}} M^{-1} \mathbf{p} = \|L_{M^{-1}}^{\mathrm{T}} \mathbf{p}_{(L)})\|^2\}$
$u \sim \mathcal{E}(u|1)$
if $r > -u$ **then**
    return $\mathbf{f}_{(L)}$
else
    return $\mathbf{f}_{(0)}$
end if

4.4 Manifold MCMC

Manifold MCMC methods (Girolami and Calderhead, 2011) were proposed to have an automatic mechanism to tune parameters in MALA and HMC, and are based on the use of curvature through the Fisher Information matrix. In this section, we provide a brief discussion on the main ideas underpinning manifold methods starting from general concepts of information geometry. Manifold methods make use of the natural geometry of the underlying statistical model to achieve efficient sampling; this idea has been presented and tested on several challenging problems in Girolami and Calderhead (2011). Key quantities in information geometry are the Fisher Information matrix (FI) and the Christoffel symbols that characterize the curvature and the connection on the manifold respectively. Consider a statistical model $S = \{p(\mathbf{y}|\boldsymbol{\psi})|\boldsymbol{\psi} \in \Psi\}$ where $\mathbf{y}$ denotes observed variables and $\boldsymbol{\psi}$ comprises all model parameters. Under conditions that are generally satisfied for most commonly used models (Amari and Nagaoka, 2000), S can be considered a C^∞ manifold, and is called statistical

manifold. Let $\mathcal{L} = \log[p(\mathbf{y}|\boldsymbol{\psi})]$; the FI matrix G of S at $\boldsymbol{\psi}$ is defined as:

$$G(\boldsymbol{\psi}) = \mathrm{E}_{p(\mathbf{y}|\boldsymbol{\psi})}\left[(\nabla_{\boldsymbol{\psi}}\mathcal{L})\,(\nabla_{\boldsymbol{\psi}}\mathcal{L})^{\mathrm{T}}\right] = -\mathrm{E}_{p(\mathbf{y}|\boldsymbol{\psi})}[\nabla_{\boldsymbol{\psi}}\nabla_{\boldsymbol{\psi}}\mathcal{L}]$$

By definition, the FI matrix is positive semidefinite:

$$\sum_{i,j} c_i c_j g_{ij} = \mathrm{E}_{p(\mathbf{y}|\boldsymbol{\psi})}\left[\sum_{i,j} c_i c_j \frac{\partial \mathcal{L}}{\partial \psi_i}\frac{\partial \mathcal{L}}{\partial \psi_j}\right] = \mathrm{E}_{p(\mathbf{y}|\boldsymbol{\psi})}\left[\sum_{i}\left(c_i \frac{\partial \mathcal{L}}{\partial \psi_i}\right)^2\right] \geq 0$$

and can be considered as the natural metric on S.

4.4.1 Fisher Information and model structure

Let's now analyze the SA case for LGMs: $p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. In this case, we immediately see that the observed variables $\mathbf{y}$ are only directly dependent on $\mathbf{f}$. This means that the statistical manifold will be of dimension $n_{\mathbf{f}}$ only. The same applies to $\mathbf{f}$, that is dependent on $\boldsymbol{\theta}$, and therefore the distributions $p(\mathbf{f}|\boldsymbol{\theta})$ live in a statistical manifold of dimension $n_{\boldsymbol{\theta}}$. If we want to consider the application of manifold methods to sample both $\mathbf{f}$ and $\boldsymbol{\theta}$, we are therefore forced to consider the two manifold separately. Note that in order to consider the joint sampling of $\mathbf{f}$ and $\boldsymbol{\theta}$, we could combine the two metrics in this form:

$$G = \begin{pmatrix} G_{\mathbf{f},\mathbf{f}} & \mathbf{0} \\ \mathbf{0} & G_{\boldsymbol{\theta},\boldsymbol{\theta}} \end{pmatrix}$$

This would form the metric for a statistical model of dimension $n_{\mathbf{f}} + n_{\boldsymbol{\theta}}$ for the observations. However, this construction is somehow artificial, as there are two statistical models, one for $\mathbf{y}$ and one for $\mathbf{f}$. Also, note that combining the two metrics results in a block diagonal matrix which does not capture the correlation between the two groups of variables(Filippone, 2011).

In the remainder of this paper we will consider L to be the lower triangular Cholesky decomposition of K, but in principle any square root of K could be used. AA, although equivalent to SA from a generative perspective, makes $\mathbf{y}$ directly dependent on the whitened variables $\boldsymbol{\nu}$ and the hyper-parameters $\boldsymbol{\theta}$, so the statistical model is of dimension $n_{\boldsymbol{\nu}} + n_{\boldsymbol{\theta}}$. The resulting metric tensor is no longer block diagonal, and therefore gives a way to characterize the correlation between latent variables and hyper-parameters.

4.4.2 Manifold MALA and Simplified Manifold MALA

The manifold MALA (MMALA) algorithm (Girolami and Calderhead, 2011) defines a Langevin diffusion with stationary distribution $p(\mathbf{f})$ on the Riemann manifold of density functions with metric tensor $G_{\mathbf{f},\mathbf{f}}$. By employing a first order Euler integrator to solve the diffusion a proposal mechanism with density $q(\mathbf{f}'|\mathbf{f}) = \mathcal{N}(\mathbf{f}'|\boldsymbol{\mu}(\mathbf{f}, \epsilon), \epsilon^2 G^{-1}_{\mathbf{f},\mathbf{f}})$ is obtained, where ϵ is the integration step size, a parameter which needs to be tuned, and the d-th component of the mean function $\boldsymbol{\mu}(\mathbf{f}, \epsilon)_d$ is

$$\boldsymbol{\mu}(\mathbf{f}, \epsilon)_d \;=\; \mathbf{f}_d + \frac{\epsilon^2}{2}\left(G^{-1}_{\mathbf{f},\mathbf{f}}\nabla_{\mathbf{f}} W(\mathbf{f})\right)_d - \epsilon^2 \sum_{i=1}^{n}\sum_{j=1}^{n}(G^{-1}_{\mathbf{f},\mathbf{f}})_{i,j}\Gamma^d_{i,j}$$

where $\Gamma^d_{i,j}$ are the Christoffel symbols of the metric in local coordinates (Kühnel, 2005).

Similarly to MALA (Roberts and Stramer, 2002), due to the discretization error introduced by the first order approximation, convergence to the stationary distribution is not guaranteed anymore

and thus the Metropolis-Hastings ratio is employed to correct this bias. Details can be found in Girolami and Calderhead (2011).

The proposal mechanism of MMALA can be interpreted as a local Gaussian approximation to the target density similar to the adaptive Metropolis-Hastings of Haario et al. (1998). In contrast to Haario et al. (1998) however, the effective covariance matrix in MMALA is the inverse of the metric tensor evaluated at the current position and no samples from the chain are required in order to estimate it, therefore avoiding the difficulties of adaptive MCMC discussed in Andrieu and Thoms (2008). The MMALA algorithm can be seen as a generalization of the original MALA (Roberts and Stramer, 2002) since, if the metric tensor $\boldsymbol{G}(\boldsymbol{\theta})$ is equal to the identity matrix corresponding to an Euclidean manifold, then the original algorithm is recovered.

In the same spirit, it is possible to extend HMC to define Hamilton's equations on the statistical manifold. This was proposed and applied in Girolami and Calderhead (2011) and called Riemann manifold Hamiltonian Monte Carlo (RM-HMC). In this work, we won't consider RM-HMC or MMALA, as they both require the derivatives of the FI that would require several expensive operations. Instead, we will consider a simplified version of MMALA (SMMALA), where we assume a manifold with constant curvature (that effectively removes the term depending on the Christoffel symbols) so that the mean of the proposal of SMMALA becomes

$$\boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon)_d = \mathbf{f}_d + \frac{\epsilon^2}{2}\left(G^{-1}_{\mathbf{f},\mathbf{f}}\nabla_{\mathbf{f}} W(\mathbf{f})\right)_d$$

The SMMALA algorithm is sketched in 4.

Algorithm 4 SMMALA

1: $\mathbf{f}' \sim \mathcal{N}(\mathbf{f}'|\boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon), \epsilon^2 G^{-1}_{\mathbf{f},\mathbf{f}})$
2: $r = \min\{0, W(\mathbf{f}') - W(\mathbf{f}) + \log[q(\mathbf{f}|\mathbf{f}')] - \log[q(\mathbf{f}'|\mathbf{f})]\}$
3: $u \sim \mathcal{E}(u|1)$
4: **if** $r > -u$ **then**
5: **return** $\mathbf{f}'$
6: **else**
7: **return** $\mathbf{f}$
8: **end if**

4.4.3 Elliptical Slice sampling - ELL-SS

Elliptical slice sampling has been proposed in Murray et al. (2010) to draw samples for $\mathbf{f}$ in LGMs. The idea underpinning ESS is based on slice sampling (Neal, 2003). Due to the fact that latent variables are Gaussian, it is possible to derive this particular version of slice sampling that is constrained in an ellipse. For completeness, we report the algorithm in Algorithm 5 and we refer the reader to Murray et al. (2010) for further details.

Note that this algorithm is quite appealing as it returns a sample which doesn't need to be accepted or rejected (in fact, a rejection mechanism is implicit within step 7), and the proposal mechanism doesn't have any free parameters that need to be tuned. Once the Cholesky factorization of K is available, the complexity of each draw using ELL-SS is in $O(n^2)$.

Algorithm 5 Elliptical Slice Sampling proposal

1: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, K)$
2: $u \sim \mathcal{E}(u|1)$
3: Set a threshold on the log-likelihood: $\eta = \log p(\mathbf{y}|\mathbf{f}) - u$
4: $\alpha \sim U[0, 2\pi]$
5: Define the bracket $[\alpha_{\min}, \alpha_{\max}] = [\alpha - 2\pi, \alpha]$
6: $\mathbf{f}' = \mathbf{f}\cos(\alpha) + \mathbf{z}\sin(\alpha)$
7: **if** $\log p(\mathbf{y}|\mathbf{f}') > \eta$ **then**
8: **return** $\mathbf{f}'$
9: **else** {shrink the bracket}
10: **if** $\alpha < 0$ **then**
11: $\alpha_{\min} = 0$
12: **else**
13: $\alpha_{\max} = 0$
14: **end if**
15: $\alpha \sim U[\alpha_{\min}, \alpha_{\max}]$
16: Go to 6
17: **end if**

5 Results on simulated data

In this section, we report an extensive study on simulated data with the aim to assess the improvement in sampling efficiency when combining different parametrizations using ASIS. Before doing so, we report a study on the efficiency of different samplers in sampling from posterior distribution of individual groups of variables, namely latent functions and hyper-parameters.

5.1 Experimental setup

We simulated data from the following models belonging to the class of LGMs: Logistic Regression with GP priors, Gaussian Copula Process Volatility model, Log-Gaussian Cox model, and Ordinal Regression with GP priors. We impose Gamma priors on the length-scale parameters:

$$p(\tau_i) = \mathcal{G}(\tau_i|a, b) \propto \tau_i^{a-1} \exp(-b\tau_i)$$

where a and b are shape and rate parameters. This corresponds to a density for each ψ_{τ_i} which is

$$p(\psi_{\tau_i}) \propto \exp(\psi_{\tau_i})^a \exp(-b \exp(\psi_{\tau_i}))$$

We impose an inverse Gamma prior on the parameter σ to exploit conjugacy in sampling σ using SA:

$$p(\sigma) = \text{inv}\mathcal{G}(\sigma|a, b) \propto \sigma^{-a-1} \exp\left(-\frac{b}{\sigma}\right)$$

The remainder of this section reports an assessment of different sampling strategies for each step of ASIS, in particular $\mathbf{f}|\mathbf{y}, \boldsymbol{\theta}$, $\boldsymbol{\theta}|\mathbf{f}$ (SA), and $\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\nu}$. We considered covariance functions with a length-scale parameter for each covariate, data sets with $n = 100, 400$, and $d = 2, 10$. Unless otherwise stated, in all the experiments on simulated data, we collected 10000 samples after a burn-in phase of 5000 iterations; during the burn-in we also had an adaptive phase to allow the samplers reach recommended acceptance rates (for example around 25% for MH).

5.2 Assessing the efficiency of samplers for individual groups of variables

In this section, we present an assessment of the efficiency of different samplers for each group of variables using both SA and AA parametrizations. The goal of this analysis is to see which samplers are most suitable to be combined into the ASIS scheme. We made extensive tests on data sets with varying number of samples and number of covariates synthetically generated from different models belonging to LGMs.

We first present the sampling of the latent variables, and then we move onto presenting results on the sampling of the hyper-parameters using SA and AA, reporting sampling efficiency with respect to computational complexity.

All implemented methods were tested for correctness as proposed by Geweke (2004), and convergence analysis was performed using the $\hat{R}$ potential scale reduction factor (Gelman and Rubin, 1992). The value of $\hat{R}$ is computed based on 10 chains initialized from the prior, rather than using the initialization procedure suggested in Gelman and Rubin (1992). This provides a tough test for convergence, and it gives an idea of what it is possible to achieve in terms of efficiency when the samplers are initialized without any preliminary runs.

In order to compare the efficiency of the samplers, we will consider the minimum of the Effective Sample Size (ESS) computed across all the sampled quantities and will report its mean and standard deviation across 10 chains. As a measure of complexity, we will evaluate the number of operations with complexity in $O(n^3)$, namely Cholesky factorizations and multiplications and inversions of $n \times n$ matrices. We believe that this is a more reliable measure of complexity with respect to running time, as running time can be affected by several factors that cannot be directly controlled by the user.

5.2.1 Sampling $\mathbf{f}|\mathbf{y}, \boldsymbol{\theta}$

In this section we focus on posterior sampling of the latent variables $\mathbf{f}$. The samplers that we tested are MH with $\Sigma = \alpha I$, HMC with mass matrix $M = \alpha I$, SMMALA with step-size ε, and ELL-SS. Preliminary runs showed that the use of gradients and curvature were beneficial for achieving high efficiency in sampling the latent variables.

We therefore tried two variants of HMC: HMC v1, and HMC v2. HMC v1 is motivated by analyzing the metric tensor for $\mathbf{f}$ which is $G_{\mathbf{f}} = K^{-1} + R$. We would like to exploit the geometric argument using RM-HMC, as it would make the sampling of $\mathbf{f}$ quite efficient. However, the complexity would be quite high, given that the metric tensor is $G_{\mathbf{f}}$ is a function of $\mathbf{f}$ in general, and we need to compute the inverse and the derivatives of $G_{\mathbf{f}}$ with respect to all elements of $\mathbf{f}$ at any implicit iteration of the leapfrog steps. A way of reducing the computational burden, while retaining a good efficiency in the proposal mechanism, is to sample $\mathbf{f}$ using HMC with a fixed mass matrix $M = G_{\mathbf{f}}$ through the leapfrog steps. We would like to set the mass matrix of HMC to be equal to $G_{\mathbf{f}}$, but we notice that R, and therefore $G_{\mathbf{f}}$ depend on the current position of $\mathbf{f}$, and this would affect detailed balance. In order to keep detailed balance satisfied, we propose to use an approximation of $G_{\mathbf{f}}$ that doesn't depend on the current position of $\mathbf{f}$. We propose to realize this by setting $M = \tilde{G}_{\mathbf{f}} = K^{-1} + \tilde{r}I$, where $\tilde{r}$ is any diagonal element of the diagonal matrix R computed with $\mathbf{f} = \mathbf{0}$ (which is the expected value of $\mathbf{f}$ a priori). By one of Woodbury identities, we can obtain a numerically stable version of the inverse of M and we can then use the variant of HMC that requires the decomposition of the inverse of M:

$$M^{-1} = \tilde{G}^{-1} = (K^{-1} + \tilde{r}I)^{-1} = \frac{1}{\tilde{r}} I - \frac{1}{\tilde{r}^2} \left(K + \frac{I}{\tilde{r}} \right)^{-1}$$

We will call this variant HMC v1.

Given that the first variant of HMC will still be expensive to compute, as it requires the inversion of an $n \times n$ matrix, we propose another variant of HMC based on $M^{-1} = K$. In this case, the Cholesky factor of M^{-1} will be the same as the Cholesky factor of K, which is needed anyway to compute the log-joint density. Again, we can make use of the formulation of HMC that requires only the Cholesky factor of the inverse of M. We will call this variant HMC v2.

The results can be found in the tables in appendix C. From these results we can observe that Ordinal Regression models behave differently than the others and this is due to the instability in computing the gradients for this model. The particular form of the likelihood makes the gradients become badly conditioned regardless of whether we use safe numerical operations or not. In this case, all samplers based on gradient and curvature information are not suitable as confirmed by the results, and ELL-SS seems to be the one offering the best performance in terms of efficiency and convergence.

For the other models, MH with identity covariance matrix and HMC with identity mass matrix do not offer guarantees of converging to the posterior distribution. SMMALA, which uses gradient and curvature information seems to offer good efficiency and convergence, but at the cost of one operation in $O(n^3)$ at each iteration. This is similar to HMC v1, that is probably the best sampler for this group of variables, but again is quite expensive. HMC v2 is competitive with HMC v1 and doesn't require any operations in $O(n^3)$ once the covariance matrix is factorized. For these reasons, we suggest HMC v2 for all models except for Ordinal Regression, where we suggest ELL-SS.

5.2.2 SA - Sampling $\boldsymbol{\theta}|\mathbf{f}$

In this section we present the sampling of the hyper-parameters from the posterior distribution of $\boldsymbol{\theta}$ given observations and latent variables using the SA parametrization. Given the hierarchical structure of the model, we immediately see that this amounts in sampling $\boldsymbol{\theta}|\mathbf{f}$. The analysis of this type of sampling is therefore independent from the particular LGM considered. Note the update of the hyper-parameters has complexity in $O(n^3)$, as the covariance needs to be updated and its Cholesky factorization needs to be recomputed.

We consider three samplers, MH, HMC, and SMMALA, with the goal of understanding if gradient and curvature information lead to any gains in the efficiency of the samplers. As it can be noticed from the tables in appendix D reporting the results for the sampling of $\boldsymbol{\theta}|\mathbf{f}$, the complexity of applying SMMALA and HMC is quite high compared to MH. HMC improves quite substantially on efficiency, but not enough if we account the computational cost. Due to these aspects, we therefore propose to use a simple MH proposal for this part of the sampling. The reason why SMMALA does not offer improvements in the efficiency might be due to the particular shape of the target density. In particular, SMMALA requires the tuning of an integration step that is used across the whole space and might not be optimal for proposals in certain parts of the parameter space. This effect due to the skewness of the target distribution was noticed in Stathopoulos and Filippone (2011) in the context of the univariate binomial probit model, and might be the reason for the inefficiency in sampling $\boldsymbol{\theta}|\mathbf{f}$ using SMMALA.

5.2.3 AA - Sampling $\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\nu}$

Again, we consider three samplers, MH, HMC, and SMMALA, with the same goal of assessing if there is any benefit in using gradient and curvature information in the sampling of $\boldsymbol{\theta}$ using the AA parametrization. In this case, we run all the tests on the four LGMs. All the results can be found in the tables in appendix E. Similarly to SA, the update of the hyper-parameters has complexity in $O(n^3)$. Again, the complexity of applying SMMALA and HMC is quite high compared to MH.

In this case, however, both SMMALA and HMC don't yield any actual gain in efficiency, and again we propose to employ a MH proposal for this part of the sampling. Similarly to the SA scheme, the reason why SMMALA and HMC do not offer improvements in the efficiency might be due to the particular shape of the target density.

5.3 Comparing ASIS with other sampling strategies

After analyzing the results reported in appendix, we decided to combine the sampler which were more promising and achieved a good efficiency with relatively low computational effort. We decided that a good combination of samplers for LGMs could be as follows: $\mathbf{f}$ using HMC v2, $\boldsymbol{\theta}$ (both SA and AA) using MH. For Ordinal Regression, we propose to sample $\mathbf{f}$ using ELL-SS, so that the proposed scheme is: $\mathbf{f}$ using ELL-SS, $\boldsymbol{\theta}$ (both SA and AA) using MH. We compared these versions of ASIS for all the models with SA and AA to see if the combination of the two schemes using ASIS improves on efficiency and convergence.

We also compared these samplers with the sampling scheme proposed in Knorr-Held and Rue (2002), that we will denote as KHR. KHR was proposed to jointly sample parameters and latent variables in applications making use of Markov Random Fields. The joint sampler proposes a set of hyper-parameters $\boldsymbol{\theta}'|\mathbf{f}, \boldsymbol{\theta}$ and then proposes a set of latent variables conditioned on the new set of hyper-parameters, namely $\mathbf{f}'|\mathbf{y}, \mathbf{f}, \boldsymbol{\theta}'$. The proposal $(\boldsymbol{\theta}', \mathbf{f}')$ is then jointly accepted or rejected according to a Metropolis acceptance ratio. In our version of KHR, we decided to sample the latent variables using ELL-SS in order to avoid difficulties in tuning proposal parameters for this group of variables. The results of this comparison are reported in appendix F for models using isotropic and diagonal covariance functions.

The results show that ASIS marginally improves convergence and efficiency with respect to the AA scheme in the data sets that we tested here. Also, the efficiency given by ASIS using HMC v2 for sampling the latent functions seems to be best in all models except for Ordinal Regression, as expected. ASIS always outperforms KHR which is a state-of-the-art algorithm that allows to sample hyper-parameters and latent variables in LGMs.

In order to assess whether ASIS is providing any actual gains in efficiency compared to AA and SA individually, we decided to carry out a further test trying to rule out the effect due to the inefficiency given by the rejection mechanism within the Gibbs sampler. In order to do so, we repeated each step of the Gibbs sampler several times thus simulating a situation in which the Gibbs steps were exact. We compared ASIS, SA, and AA on the four models on simulated data sets with increasing number of samples $n = 10, 100, 1000$, and in two dimensions. The reason for such an experimental setting is to see the effect of combining AA and SA into ASIS in the strong/weak data limits.

The results of this tests are reported in Tabs. 1–4, where we report the minimum ESS across all the variables (both $\mathbf{f}$ and $\boldsymbol{\theta}$), its standard deviation (in parenthesis) computed on the basis of 10 runs, and the value of $\hat{R}$ again based on 10 runs. The results reported in these tables are based on 5000 samples obtained after 5000 burn-in samples. The general trend is that the efficiency of the schemes decrease in the strong data limit. The results show that in general ASIS provides gains in speed of convergence compared to the SA and AA schemes, especially for large data sets, but the efficiency is generally comparable to the AA scheme. In the case of log-Gaussian Cox model, instead, the efficiency of ASIS is superior to the AA scheme, although it might not be computationally advantageous to employ ASIS as the improvement is less or of the order than double the efficiency of AA or SA. The fact that we are studying models where the covariance of the latent variables is full, prevents us to work with data sets larger than a couple of thousands samples, so we can deduce that in the case

considered here the AA parametrization will be equally preferable to ASIS given that ASIS requires double the number of operations in $O(n^3)$. SA is usually less preferable than AA, as already pointed out previously and as shown in the results.

Table 1: Logistic regression – $d = 2$

Sampler	$n = 10$	$n = 100$	$n = 1000$
ASIS	2328 (49), 1.00	465 (63), 1.00	130 (34), 1.01
SA	132 (34), 1.01	28 (4), 1.05	21 (1), 1.17
AA	2347 (69), 1.00	537 (63), 1.00	128 (40), 1.02

Table 2: Log-Gaussian Cox model – $d = 2$

Sampler	$n = 10$	$n = 100$	$n = 1000$
ASIS	1791 (69), 1.00	191 (52), 1.01	41 (5), 1.02
SA	246 (32), 1.00	33 (4), 1.04	20 (1), 1.21
AA	1593 (109), 1.00	156 (13), 1.01	37 (9), 1.06

Table 3: Gaussian Copula Process Volatility model – $d = 2$

Sampler	$n = 10$	$n = 100$	$n = 1000$
ASIS	1640 (67), 1.00	175 (25), 1.01	31 (6), 1.03
SA	764 (47), 1.00	41 (10), 1.02	21 (2), 1.20
AA	767 (67), 1.00	103 (13), 1.02	31 (2), 1.10

6 Results on Real Data

In this section we report results on real data sets and comparisons with another state-of-the-art sampling method that was recently presented in Murray and Adams (2010). This method, that we will refer to as the Surrogate method, makes use of efficient parametrization techniques based on surrogate data points. The strategy proposed in Murray and Adams (2010) amounts in using 10 iterations of ELL-SS for sampling the latent variables and component-wise slice sampling for updating the hyper-parameters using a parametrization based on surrogate data.

We report a comparison of efficiency and number of operations in $O(n^3)$ on six UCI data sets (Asuncion and Newman, 2007): Pima, Wisconsin, SPECT, Ionosphere, Mining, and Redwood. The type of LGM and number of samples and number of covariates used for inferring all quantities are reported in Tab. 5. In the case of logistic regression we selected a subset of the whole data set for inferring parameters and latent variables so that for the Ionosphere data set we could directly compare the performance of AA with the method proposed in Murray and Adams (2010). Also, we report this

Table 4: Ordinal regression with GP prior – $d = 2$

Sampler	$n = 10$	$n = 100$	$n = 1000$
ASIS	1640 (109), 1.00	98 (14), 1.01	21 (1), 1.12
SA	1323 (157), 1.00	28 (3), 1.13	19 (1), 1.29
AA	301 (34), 1.00	45 (11), 1.02	24 (4), 1.19

Table 5: Number of samples used for inference and dimensionality of the considered real data sets.

Data	# samples	d	LGM
Pima	200	8	logistic regression
Wisconsin	100	10	logistic regression
SPECT	80	22	logistic regression
Ionosphere	200	34	logistic regression
Mining	112	1	log-Gaussian Cox
Redwood	625	2	log-Gaussian Cox

comparison on two LGMs only, as we can use directly the code implementing the Surrogate method as distributed by Murray and Adams (2010) that can be applied to logistic regression and log-Gaussian Cox models.

We ran our chains for 100000 iterations and used the first 50000 for burn-in and the 50000 samples after the burn-in to evaluate the ESS. We kept the same experimental setting as in Murray and Adams (2010), whereby uniform priors where assigned to the hyper-parameters in the range $[0.01, 10]$ (before log-transforming). Also, in order to keep the same experimental setting, we employed ELL-SS for sampling $\mathbf{f}$ and each proposal comprised 10 iterations of ELL-SS. We compare the Surrogate method with the AA parametrization as we expect to yield similar results to ASIS. In order to roughly match the complexity of the Surrogate method, we repeated the sampling of the hyper-parameters 40 times within each Gibbs step in the AA parametrization and 80 times in the case of the Ionosphere data set. In the case of log-Gaussian Cox models, we added a further Gibbs step sampling a mean offset to the log-rate with a uniform prior in $[-10, 10]$ as in Murray and Adams (2010).

The results are reported in Tab. 6 and show how with the same number of operations in $O(n^3)$ with respect to the Surrogate method, or even less, it is possible to achieve substantially higher sampling efficiency. This is mainly due to the fact that the Surrogate methods employs an expensive component-wise update of the hyper-parameters that doesn't exploit joint updates as the MH algorithm. For logistic regression, the ESS is generally around a few percent of the number of MCMC samples gathered after the burn-in phase. In the case of the Wisconsin data set, instead, the proposed scheme reaches an ESS which is around 28%, which is quite remarkable given the challenging nature of the problem. For the log-Gaussian Cox model, the results are slightly worse, and this might be due to the fact that there is a further Gibbs step that reduces the overall efficiency.

Table 6: UCI data sets - comparison of the proposed sampling scheme with the Surrogate method.

Data	Method	$10^6 \# O(n^3)$	10^2 ESS
Pima	AA	4.0	11.3
	Surrogate	4.4	0.6
Wisconsin	AA	4.0	141.0
	Surrogate	10.7	6.3
SPECT	AA	4.0	16.7
	Surrogate	12.8	2.8
Ionosphere	AA	8.0	13.5
	Surrogate	30.0	5.3
Mining	AA	2.0	2.9
	Surrogate	1.2	0.25
Redwood	AA	2.0	4.0
	Surrogate	2.9	1.18

7 Conclusions

In this paper we studied and compared a number of strategies to efficiently carry out the fully Bayesian treatment of Latent Gaussian Models (LGMs), which is an extremely challenging problem. We focused on four LGMs, namely Logistic Regression with GP priors, Gaussian Copula Process Volatility model, Log-Gaussian Cox model, and Ordinal Regression with GP priors and we carried out extensive evaluations of several sampling methods and strategies on simulated and real data. We discussed how the Metropolis-within-Gibbs samplers based on two different parametrizations can be derived and we studied empirically which samplers provided the best performance in terms of convergence and efficiency for each step of the Gibbs samplers. In particular, we compared several sampling algorithms based on different degrees of complexity such as MH which is based on a random walk type of proposal, HMC which exploits gradient information, and manifold methods that make use of curvature information.

The results of extensive tests showed that it is possible to sample the latent variables using a simple modification of the HMC algorithm where the mass matrix is equal to the inverse of the covariance matrix. The derivation reported in this paper, shows that once the Cholesky factor of the covariance is available, drawing samples from the posterior distribution of the latent variables is quite efficient and has complexity in $O(n^2)$.

When sampling hyper-parameters, the results showed that gains in efficiency due to the use of complicated proposal mechanisms are usually overwhelmed by the complexity of the proposal mechanism itself. We therefore suggest to employ a simple random walk mechanism for which each proposal costs only one operation in $O(n^3)$ for both SA and AA parametrizations.

We then compared AA and SA parametrization and their combination through interweaving (ASIS) as suggested in Yu and Meng (2011), with the aim of improving efficiency and achieve faster convergence. The results of this study showed that for the considered models and scenarios it is generally efficient enough to sample using only the AA parametrization, and the combination of SA and AA schemes using ASIS is preferrable for larger data sets. The gains in efficiency and convergence given by ASIS are most of the times compensated by the cost of double updates of the

hyper-parameters. In other words, for the cases reported here, ASIS costs twice as much in terms of operations in $O(n^3)$ but usually yields an efficiency which is less than double AA's.

The results presented in this work apply to data sets with a number of samples that doesn't exceed a couple of thousands, as the time complexity of samplers updating the hyper-parameters scales with the cube of the number of samples. It is worth stressing that this scaling in complexity applies also to any other method that seeks to optimize the hyper-parameters through type II maximum likelihood (Bishop, 2007) or integrate out the hyper-parameters. Work in the direction of reducing the computational complexity of models based on GP priors includes the use of sparse structures for the latent variables (Rue et al., 2009), methods for selecting sets of relevant data on which to focus (Lawrence et al., 2002; Snelson and Ghahramani, 2005), or approximate likelihoods (Varin et al., 2011). The aim of this paper was to provide an extensive evaluation of the fully Bayesian treatment of LGMs in cases of small to moderate sample sized data sets, with a full covariance structure of the GP prior, and where the number of hyper-parameters prevents the use of quadrature methods.

References

S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical monographs*. Oxford University Press, 2000.

C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, 2008.

A. Asuncion and D. J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/$sim$mlearn/MLRepository.html`.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, August 2007.

W. Chu and Z. Ghahramani. Gaussian Processes for Ordinal Regression. *J. Mach. Learn. Res.*, 6: 1019–1041, December 2005.

B. Cseke and T. Heskes. Approximate Marginals in Latent Gaussian Models. *Journal of Machine Learning Research*, 12:417–454, 2011.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

M. Filippone. Discussion of the paper "Riemann manifold Langevin and Hamiltonian Monte Carlo methods" by Mark Girolami and Ben Calderhead. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 73(2):123–214, 2011.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

J. Geweke. Getting It Right: Joint Distribution Tests of Posterior Simulators. *Journal of the American Statistical Association*, 99(467), 2004.

W. R. Gilks and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

G. H. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition, October 1996.

H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7:223–242, 1998.

H. Haario, E. Saksman, and E. Tamminen. Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20:265–273, 2005.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

L. Knorr-Held and H. Rue. On Block Updating in Markov Random Field Models for Disease Mapping. *Scandinavian Journal of Statistics*, 29(4):597–614, December 2002.

W. Kühnel. *Differential Geometry: Curves - Surfaces - Manifolds*, volume 16 of *Student Mathematical Library*. AMS, 2005.

M. Kuss and C. E. Rasmussen. Assessing Approximate Inference for Binary Gaussian Process Classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

N. D. Lawrence, M. Seeger, and R. Herbrich. Fast Sparse Gaussian Process Methods: The Informative Vector Machine. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *NIPS*, pages 609–616. MIT Press, 2002.

D. J. C. Mackay. Bayesian methods for backpropagation networks. In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks III*, chapter 6, pages 211–254. Springer, 1994.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Uncertainty in artificial intelligence: proceedings of the seventeenth conference (2001), August 2-5, 2001, University of Washington, Seattle, Washington*, page 362. Morgan Kaufmann, 2001.

I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1723–1731. 2010.

I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

R. M. Neal. Regression and classification using Gaussian process priors (with discussion). *Bayesian Statistics*, 6:475–501, 1999.

R. M. Neal. MCMC using Hamiltonian dynamics. in *Handbook of Markov Chain Monte Carlo* (eds S. Brooks, A. Gelman, G. Jones, XL Meng). Chapman and Hall/CRC Press, 2010.

R. M. Neal. Slice Sampling. *Annals of Statistics*, 31:705–767, 2003.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, September 1993.

R. M. Neal. *Bayesian Learning for Neural Networks (Lecture Notes in Statistics)*. Springer, 1 edition, August 1996.

J. A. Nelder and R. W. M. Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3), 1972.

O. Papaspiliopoulos, G. O. Roberts, and M. Sköld. A General Framework for the Parametrization of Hierarchical Models. *Statistical Science*, 22(1), 2007.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

G. O. Roberts and O. Stramer. Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodology and Computing in Applied Probability*, 4(4):337–357, December 2002.

G. O. Roberts and J. S. Rosenthal. Optimal Scaling of Discrete Approximations to Langevin Diffusions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(1):255–268, 1998.

G. O. Roberts and J. S. Rosenthal. Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statistical Science*, 16(4):351–367, 2001.

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

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *NIPS*, 2005.

V. Stathopoulos and M. Filippone. Discussion of the paper "Riemann manifold Langevin and Hamiltonian Monte Carlo methods" by Mark Girolami and Ben Calderhead. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 73(2):123–214, 2011.

L. Tierney and J. B. Kadane. Accurate Approximations for Posterior Moments and Marginal Densities. *Journal of the American Statistical Association*, 81(393), 1986.

D. van Dyk and X. L. Meng. The art of data augmentation (with discussion). *Journal of Computational and Graphical Statistics*, 10:1–111, 2001.

C. Varin, N. Reid, and D. Firth. An Overview of Composite Likelihood Methods. *Statistica Sinica*, 21:5–42, 2011.

Y. Yu and X. L. Meng. To Center or Not to Center, That is Not the Question: An Ancillarity-Sufficiency Interweaving Strategy (ASIS) for Boosting MCMC Efficiency. *Journal of Computational and Graphical Statistics*, to appear, 2011.

A ASIS for LGMs

A.1 Sufficient Augmentation (SA)

In this section we derive the key quantities needed to apply manifold and Hamiltonian MCMC based methods in the SA parametrization.

A.1.1 Log-joint density

Let $\mathcal{L} = \log[p(\mathbf{y}|\mathbf{f})]$ The log-joint density is:

$$\log[p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta})] = \mathcal{L} - \frac{1}{2}\log(|Q|) - \frac{n}{2}\log(\sigma) - \frac{1}{2\sigma}\mathbf{f}^{\mathrm{T}}Q^{-1}\mathbf{f} + \log[p(\boldsymbol{\theta})] + \text{const.}$$

Note that we could marginalize out σ, but it wouldn't be possible to get manageable expressions for the metric tensor w.r.t $\boldsymbol{\tau}$; for $\mathbf{f}$, instead, it would be possible.

A.1.2 Gibbs sampling σ

By inspecting the log-joint density, we see that we can obtain the marginal for σ in closed form

$$\log[p(\sigma|\mathbf{y}, \mathbf{f}, \boldsymbol{\tau})] = -\frac{n}{2}\log(\sigma) - \frac{1}{2\sigma}\mathbf{f}^{\mathrm{T}}Q^{-1}\mathbf{f} + \text{const.}$$

which we recognize as an inverse Gamma. By placing an inverse Gamma prior $\text{invGa}(a, b)$ on σ, we can sample:

$$\sigma \sim \text{invGa}\left(\sigma \middle| a + \frac{n}{2}, b + \frac{1}{2\sigma}\mathbf{f}^{\mathrm{T}}Q^{-1}\mathbf{f}\right)$$

A.1.3 Gradients of the Log-joint density

$$\nabla_{\mathbf{f}} \log[p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta})] = \nabla_{\mathbf{f}}\mathcal{L} - \frac{1}{\sigma}Q^{-1}\mathbf{f}$$

$$\frac{\partial \log[p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta})]}{\partial \psi_{\tau_i}} = \frac{\partial \log[p(\mathbf{f}|\boldsymbol{\theta})]}{\partial \psi_{\tau_i}} + \frac{\partial \log[p(\boldsymbol{\psi}_{\boldsymbol{\tau}})]}{\partial \psi_{\tau_i}}$$

$$\frac{\partial \log[p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta})]}{\partial \psi_{\tau_i}} = -\frac{1}{2}\text{Tr}\left(Q^{-1}\frac{\partial Q}{\partial \psi_{\tau_i}}\right) + \frac{1}{2\sigma}\mathbf{f}^{\mathrm{T}}Q^{-1}\frac{\partial Q}{\partial \psi_{\tau_i}}Q^{-1}\mathbf{f} + \frac{\partial \log[p(\boldsymbol{\psi}_{\boldsymbol{\tau}})]}{\partial \psi_{\tau_i}}$$

A.1.4 Fisher Information and metric tensors

The FI for latent functions and parameters are:

$$\mathrm{FI}_{\mathbf{f},\mathbf{f}} = \mathrm{E}_{\mathbf{y}}\left[(\nabla_{\mathbf{f}}\mathcal{L})(\nabla_{\mathbf{f}}\mathcal{L})^{\mathrm{T}}\right]$$

$$\mathrm{FI}_{\boldsymbol{\psi}_{\boldsymbol{\tau}},\boldsymbol{\psi}_{\boldsymbol{\tau}}} = \mathrm{E}_{\mathbf{f}}\left[(\nabla_{\boldsymbol{\psi}_{\boldsymbol{\tau}}}\log[p(\mathbf{f}|\boldsymbol{\psi}_{\boldsymbol{\tau}})])(\nabla_{\boldsymbol{\psi}_{\boldsymbol{\tau}}}\log[p(\mathbf{f}|\boldsymbol{\psi}_{\boldsymbol{\tau}})])^{\mathrm{T}}\right]$$

Define the matrix:

$$R = \mathrm{E}_{\mathbf{y}}\left[(\nabla_{\mathbf{f}}\mathcal{L})(\nabla_{\mathbf{f}}\mathcal{L})^{\mathrm{T}}\right]$$

The metric tensors are the FI plus the negative Hessian of the prior:

$$G_{\mathbf{f},\mathbf{f}} = R + \frac{1}{\sigma}Q^{-1}$$

$$G_{\psi_{\tau_i},\psi_{\tau_j}} = +\frac{1}{2}\mathrm{Tr}\left(Q^{-1}\frac{\partial Q}{\partial \psi_{\tau_j}}Q^{-1}\frac{\partial Q}{\partial \psi_{\tau_i}}\right) - \frac{\partial^2 \log[p(\boldsymbol{\psi}_{\boldsymbol{\tau}})]}{\partial \psi_{\tau_i}\partial \psi_{\tau_j}}$$

A.2 Ancillary Augmentation (AA)

In this section, like in the former section, we derive the key quantities needed to apply manifold and Hamiltonian MCMC based methods in the AA parametrization.

A.2.1 Log-joint density

The expression of the likelihood for the observations is the same as in the SA case, bearing in mind the transformation:

$$\mathbf{f} = \sqrt{\sigma}L\boldsymbol{\nu}$$

The log-joint density is:

$$\log[p(\mathbf{y},\boldsymbol{\nu},\boldsymbol{\theta})] = \mathcal{L}(\mathbf{y}|\boldsymbol{\nu},\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{\nu} + \log[p(\boldsymbol{\theta})] + \mathrm{const.}$$

A.2.2 Gradients of the log-joint density

The gradients can be computed by using the chain rules of derivation and standard properties of derivatives of vector valued functions. In particular, we can easily obtain the expressions for the gradients with respect to $\boldsymbol{\nu}$ by noticing that the Jacobian of the transformation is $\sqrt{\sigma}L^{\mathrm{T}}$.

$$\nabla_{\boldsymbol{\nu}}\log[p(\mathbf{y},\boldsymbol{\nu},\boldsymbol{\theta})] = \sqrt{\sigma}L^{\mathrm{T}}\nabla_{\mathbf{f}}\mathcal{L} - \boldsymbol{\nu}$$

$$\frac{\partial \log[p(\mathbf{y},\boldsymbol{\nu},\boldsymbol{\theta})]}{\partial \psi_{\tau_i}} = \sqrt{\sigma}(\nabla_{\mathbf{f}}\mathcal{L}(\mathbf{y}|\mathbf{f}))^{\mathrm{T}}\frac{\partial L}{\partial \theta_i}\boldsymbol{\nu} + \frac{\partial \log[p(\boldsymbol{\psi}_{\boldsymbol{\tau}})]}{\partial \psi_{\tau_i}}$$

A.2.3 Fisher Information and Metric tensor

The expectation taken with respect to $\mathbf{y}$ of the outer product of the score functions gives the FI:

$$\mathrm{FI}_{\boldsymbol{\nu},\boldsymbol{\nu}} = \sigma L^{\mathrm{T}} R L$$

$$\mathrm{FI}_{\boldsymbol{\nu},\theta_j} = \sigma L^{\mathrm{T}} R \frac{\partial L}{\partial \theta_i} \boldsymbol{\nu}$$

$$\mathrm{FI}_{\theta_i,\theta_j} = \sigma \boldsymbol{\nu}^{\mathrm{T}} \frac{\partial L^{\mathrm{T}}}{\partial \theta_i} R \frac{\partial L}{\partial \theta_j} \boldsymbol{\nu}$$

With the contribution (negative Hessian) of the prior, the metric tensor used in the manifold methods results in:

$$G_{\boldsymbol{\nu},\boldsymbol{\nu}} = \sigma L^{\mathrm{T}} R L + I$$

$$G_{\boldsymbol{\nu},\theta_i} = \sigma L^{\mathrm{T}} R \frac{\partial L}{\partial \theta_i} \boldsymbol{\nu}$$

$$G_{\theta_i,\theta_j} = \sigma \boldsymbol{\nu}^{\mathrm{T}} \frac{\partial L^{\mathrm{T}}}{\partial \theta_i} R \frac{\partial L}{\partial \theta_j} \boldsymbol{\nu} - \frac{\partial^2 \log[p(\boldsymbol{\theta})]}{\partial \theta_i \partial \theta_j}$$

$$G = \begin{pmatrix} G_{\boldsymbol{\nu},\boldsymbol{\nu}} & G_{\boldsymbol{\nu},\boldsymbol{\theta}} \\ G_{\boldsymbol{\theta},\boldsymbol{\nu}} & G_{\boldsymbol{\theta},\boldsymbol{\theta}} \end{pmatrix}$$

The derivatives of G follow from standard properties of matrix derivatives (see the Appendix for details). We note here that in the AA, G turns out to be full, so this gives a geometric argument that can be exploited in sampling in LGMs.

A.2.4 Structure of the FI

In this section we analyze the likelihood $\mathcal{L}$:

$$\mathcal{L} = \sum_{i=1}^{n} \log[h(y_i)] + \sum_{i=1}^{n} \log\{g[\zeta(f_i)]\} + \sum_{i=1}^{n} \zeta(f_i) u(y_i)$$

Its gradient is:

$$(\nabla_{\mathbf{f}} \mathcal{L})_j = \frac{\partial \zeta(f_j)}{\partial f_j} \left(\frac{1}{g[\zeta(f_j)]} \frac{\partial g[\zeta(f_j)]}{\partial \zeta(f_j)} + u(y_j) \right)$$

We recall that:

$$\mathrm{E}_{\mathbf{y}} \left[(\nabla_{\mathbf{f}} \mathcal{L})(\nabla_{\mathbf{f}} \mathcal{L})^{\mathrm{T}} \right] = -\mathrm{E}_{\mathbf{y}} \left[\nabla_{\mathbf{f}} \nabla_{\mathbf{f}} \mathcal{L} \right]$$

Therefore, given that the Hessian with respect to $\mathbf{f}$ is diagonal, the FI will be diagonal as well.

B Special cases of LGMs considered in this paper

B.1 Logistic regression with GP priors

Let:

$$l^+(f) = \text{logistic}(f) = \frac{1}{1+\exp(-f)}$$

and

$$l^-(f) = 1 - l^+(f) = \text{logistic}(-f)$$

In logistic regression, the observations follow a Bernoulli distribution with success probability given by a sigmoid transformation of the associated latent variable:

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} \text{Bern}(y_i|l^+(f_i)) = \prod_{i=1}^{n} l^+(f_i)^{y_i} l^-(f_i)^{(1-y_i)}$$

The Bernoulli distribution is a member of the exponential family of distributions[2]. In the logistic regression case, we identify:

$$\zeta(f_i) = \log\left(\frac{l^+(f_i)}{l^-(f_i)}\right) = f_i$$

$$u(y_i) = y_i$$

$$g[\zeta(f_i)] = g[f_i] = l^-(f_i)$$

$$\frac{\partial g[\zeta(f_j)]}{\partial \zeta(f_j)} = -l^-(f_i)l^+(f_i)$$

Substituting these expressions in the expression of the gradient of $\mathcal{L}$, we obtain:

$$(\nabla_{\mathbf{f}}\mathcal{L})_j = y_j - l^+(f_j)$$

The diagonal elements of the FI for $\mathbf{f}$ need the expectations of y_i^2 which is the same as the expectation of y_i, that is l_i^+. We easily find that:

$$R_{ii} = l^+(f_i)l^-(f_i)$$

For the derivatives of the metric tensor, we need to compute $\frac{\partial R}{\partial \nu_k}$ and $\frac{\partial R}{\partial \theta_k}$.

B.2 Log-Gaussian Cox model

In this model, the observations follow a Poisson distribution with mean computed as an exponentially transformed version of the the latent variable at their location:

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} \text{Poisson}(y_i|\exp(f_i))$$

The Poisson distribution is a member of the exponential family of distributions[3].

In the log-Gaussian Cox model, we identify:

$$\zeta(f_i) = \log(\exp(f_i)) = f_i$$

[2] If $p(y|\eta) = \mathcal{E}(y|\eta) = \text{Bern}(y|\mu)$, then $\eta = \log(\frac{\mu}{1-\mu})$, $u(y) = y$, $g(\eta) = \sigma(-\eta)$, and $h(y) = 1$.

[3] If $p(y|\eta) = \mathcal{E}(y|\eta) = \text{Poisson}(y|\lambda)$, then $\eta = \log(\lambda)$, $u(y) = y$, $g(\eta) = \exp(-\exp(\eta))$, and $h(y) = 1/y!$.

$$u(y_i) = y_i$$
$$g[\zeta(f_i)] = g[f_i] = \exp(-\exp(f_i))$$
$$\frac{\partial g[\zeta(f_j)]}{\partial \zeta(f_j)} = -g[f_j]\exp(f_j)$$

Substituting these expressions in the expression of the gradient of $\mathcal{L}$, we obtain:

$$\left(\nabla_{\mathbf{f}}\mathcal{L}\right)_j = y_j - \exp(f_j)$$
$$R_{ii} = \exp(f_i)$$

B.3 Gaussian Copula Process Volatility model

In this model, the observations follow a zero mean Gaussian distribution with standard deviation computed as an exponentially transformed version of the the latent variable at their location:

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} \mathcal{N}(y_i|0, \exp(f_i)^2)$$

The Gaussian distribution is a member of the exponential family of distributions[4].

In the Gaussian Copula Process Volatility model, we identify:

$$\zeta(f_i) = -\frac{1}{2[\exp(f_i)]^2}$$
$$\frac{\partial \zeta(f_i)}{\partial f_i} = \exp(f_i)^{-2}$$
$$u(y_i) = y_i^2$$
$$g[\zeta(f_i)] = (-2\zeta(f_i))^{1/2}$$
$$\frac{\partial g[\zeta(f_j)]}{\partial \zeta(f_j)} = -\frac{1}{g[\zeta(f_i)]}$$

Substituting these expressions in the expression of the gradient of $\mathcal{L}$, we obtain:

$$\left(\nabla_{\mathbf{f}}\mathcal{L}\right)_j = \exp(f_i)^{-2} y_j^2 - 1$$
$$R_{ii} = 2$$

[4] If $p(y|\sigma) = \mathcal{E}(y|\eta) = \mathcal{N}(y|0, \sigma^2)$, then $\eta = -\frac{1}{2\sigma^2}$, $u(y) = y^2$, $g(\eta) = (-2\eta)^{1/2}$, and $h(y) = (2\pi)^{-1/2}$.

B.4 Ordinal Regression with GP priors

In this model, the latent function is thresholded at r points that will be denoted by $b_0, \ldots, b_r$ (Chu and Ghahramani, 2005). First, let $b_0 = -\infty$ and $b_r = +\infty$. Then, y is the index of the interval where the latent function f falls. The likelihood of an observed label y_i associated to a latent function value f_i is then:

$$\bar{p}(y_i|f_i) = 1 \qquad \text{if } b_{y_i-1} < f \leq b_{y_i}$$

and zero otherwise. This model is usually modified to allow for a noise term δ (distributed as $\mathcal{N}(\delta|0,\sigma_\delta^2)$) in the latent function so that:

$$p(y_i|f_i) = \int \bar{p}(y_i|f_i+\delta)\mathcal{N}(\delta|0,\sigma_\delta^2)d\delta = \Phi(z_i^{(y_i)}) - \Phi(z_i^{(y_i-1)})$$

where:

$$z_i^{(s)} = \frac{b_s - f_i}{\sigma_\delta}$$

This likelihood function does not fall in the category of exponential family, so we will derive the relevant quantities starting from the likelihood. In particular:

$$\mathcal{L} = \sum_{i=1}^{n} \log\left[\Phi(z_i^{(y_i)}) - \Phi(z_i^{(y_i-1)})\right]$$

$$(\nabla_{\mathbf{f}}\mathcal{L})_i = \frac{1}{\sigma_\delta}\frac{\mathcal{N}(z_i^{(y_i-1)}|0,1) - \mathcal{N}(z_i^{(y_i)}|0,1)}{\Phi(z_i^{(y_i)}) - \Phi(z_i^{(y_i-1)})}$$

Define:

$$(\nabla_{\mathbf{f}}\nabla_{\mathbf{f}}\mathcal{L})_{ii}^{(s)} = \frac{1}{\sigma_\delta^2}\frac{z_i^{(s)}\mathcal{N}(z_i^{(s)}|0,1) - z_i^{(s-1)}\mathcal{N}(z_i^{(s-1)}|0,1)}{\Phi(z_i^{(s)}) - \Phi(z_i^{(s-1)})} - \frac{1}{\sigma_\delta^2}\left(\frac{\mathcal{N}(z_i^{(s-1)}|0,1) - \mathcal{N}(z_i^{(s)}|0,1)}{\Phi(z_i^{(s)}) - \Phi(z_i^{(s-1)})}\right)^2$$

which is the Hessian of the likelihood computed for $y_i = s$. The off-diagonal elements of the Hessian are zero. The expectation of the negative Hessian can be computed explicitly as:

$$R_{ii} = -\sum_{s=1}^{r}(\nabla_{\mathbf{f}}\nabla_{\mathbf{f}}\mathcal{L})_{ii}^{(s)}\, p(s|f_i) = \sum_{s=1}^{r}(\nabla_{\mathbf{f}}\nabla_{\mathbf{f}}\mathcal{L})_{ii}^{(s)}\left[\Phi(z_i^{(s-1)}) - \Phi(z_i^{(s)})\right]$$

C Results - efficiency in sampling $\mathbf{f}|\mathbf{y},\boldsymbol{\theta}$

Table 7: Number of operations in $O(n^3)$

Sampler	$\#O(n^3)$
MH	1 (0)
HMC	1 (0)
SMMALA	15000 (0)
ELL_SS	1 (0)
HMC v1	15000 (0)
HMC v2	1 (0)

C.1 Logistic regression

Table 8: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 482.10
HMC	32 (0), 81.36
SMMALA	674 (56), 1.00
ELLIPTICALSS	254 (23), 1.00
HMC v1	2112 (206), 1.00
HMC v2	1559 (158), 1.00

Table 9: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	43 (1), 1.05
HMC	2202 (106), 1.00
SMMALA	244 (38), 1.00
ELLIPTICALSS	110 (6), 1.01
HMC v1	2436 (66), 1.00
HMC v2	2405 (154), 1.00

Table 10: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 1374.52
HMC	32 (0), 213.26
SMMALA	387 (28), 1.00
ELLIPTICALSS	85 (7), 1.01
HMC v1	993 (130), 1.00
HMC v2	465 (43), 1.00

Table 11: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	33 (0), 1.24
HMC	1136 (68), 1.00
SMMALA	55 (3), 1.03
ELLIPTICALSS	46 (2), 1.04
HMC v1	1328 (123), 1.00
HMC v2	1334 (100), 1.00

C.2 Log-Gaussian Cox model

Table 12: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 422.72
HMC	32 (0), 55.30
SMMALA	376 (38), 1.00
ELLIPTICALSS	69 (6), 1.02
HMC v1	695 (63), 1.00
HMC v2	240 (29), 1.00

Table 13: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	34 (0), 1.20
HMC	146 (13), 1.01
SMMALA	41 (1), 1.04
ELLIPTICALSS	36 (1), 1.09
HMC v1	145 (10), 1.01
HMC v2	146 (10), 1.01

Table 14: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 1310.06
HMC	32 (0), 127.56
SMMALA	296 (29), 1.00
ELLIPTICALSS	47 (3), 1.06
HMC v1	286 (58), 1.00
HMC v2	101 (15), 1.01

Table 15: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 1.80
HMC	98 (9), 1.01
SMMALA	33 (0), 1.27
ELLIPTICALSS	32 (0), 1.68
HMC v1	98 (10), 1.01
HMC v2	96 (9), 1.01

C.3 Gaussian Copula Process Volatility model

Table 16: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 437.49
HMC	32 (0), 74.69
SMMALA	418 (56), 1.00
ELLIPTICALSS	59 (4), 1.02
HMC v1	1489 (119), 1.00
HMC v2	183 (19), 1.00

Table 17: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	34 (1), 1.14
HMC	227 (26), 1.00
SMMALA	74 (8), 1.02
ELLIPTICALSS	36 (1), 1.12
HMC v1	242 (22), 1.00
HMC v2	236 (25), 1.00

Table 18: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 1103.14
HMC	32 (0), 172.78
SMMALA	308 (26), 1.00
ELLIPTICALSS	42 (3), 1.17
HMC v1	1057 (77), 1.00
HMC v2	80 (8), 1.01

Table 19: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 1.78
HMC	136 (15), 1.01
SMMALA	52 (2), 1.02
ELLIPTICALSS	32 (0), 1.36
HMC v1	142 (17), 1.01
HMC v2	128 (13), 1.01

C.4 Ordinal regression with GP priors

Table 20: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 373.75
HMC	32 (0), 124.05
SMMALA	35 (7), 32.14
ELLIPTICALSS	47 (3), 1.08
HMC v1	519 (785), 28.00
HMC v2	58 (34), 22.75

Table 21: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	33 (1), 2.33
HMC	32 (0), 675.48
SMMALA	32 (0), 224.54
ELLIPTICALSS	34 (1), 1.17
HMC v1	32 (0), 672.34
HMC v2	32 (0), 716.33

Table 22: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 1365.70
HMC	32 (0), 661.55
SMMALA	34 (6), 62.58
ELLIPTICALSS	40 (2), 1.37
HMC v1	32 (0), 84.51
HMC v2	32 (0), 50.62

Table 23: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$
MH	32 (0), 6.81
HMC	32 (0), 1298.09
SMMALA	32 (0), 338.10
ELLIPTICALSS	32 (0), 3.61
HMC v1	32 (0), 1338.51
HMC v2	32 (0), 1421.43

D Results - efficiency in sampling $\boldsymbol{\theta}|\mathbf{f}$ - SA

Table 24: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	775 (55), 1.00	15 (0)
HMC	3330 (142), 1.00	164 (1)
SMMALA	1900 (671), 19.37	60 (0)

Table 25: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	43 (4), 1.06	15 (0)
HMC	83 (16), 1.01	165 (1)
SMMALA	44 (5), 1.13	180 (0)

Table 26: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	1014 (118), 1.00	15 (0)
HMC	5552 (481), 1.00	164 (1)
SMMALA	2802 (2046), 14.00	60 (0)

Table 27: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	41 (3), 1.10	15 (0)
HMC	63 (11), 1.02	165 (1)
SMMALA	42 (3), 1.08	180 (0)

E Results - efficiency in sampling $\boldsymbol{\theta}|\mathrm{y}, \boldsymbol{\nu}$ - AA

E.1 Logistic regression

Table 28: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	326 (70), 1.00	15 (0)
HMC	49 (21), 1.24	660 (3)
SMMALA	80 (31), 1.16	210 (0)

Table 29: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	48 (4), 1.02	15 (0)
HMC	33 (1), 2.16	2642 (8)
SMMALA	33 (1), 1.46	2130 (0)

Table 30: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	303 (39), 1.00	15 (0)
HMC	54 (27), 3.33	660 (3)
SMMALA	87 (58), 1.43	210 (0)

Table 31: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	40 (5), 1.31	15 (0)
HMC	32 (0), 3.63	2642 (8)
SMMALA	33 (1), 2.19	2130 (0)

E.2 Log-Gaussian Cox model

Table 32: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	237 (30), 1.00	15 (0)
HMC	51 (26), 7.82	660 (3)
SMMALA	82 (65), 1.82	210 (0)

Table 33: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	37 (3), 2.11	15 (0)
HMC	32 (0), 7.20	2642 (8)
SMMALA	33 (0), 5.72	2130 (0)

Table 34: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	232 (48), 1.00	15 (0)
HMC	44 (18), 12.49	660 (3)
SMMALA	66 (49), 4.32	210 (0)

Table 35: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	36 (2), 2.43	15 (0)
HMC	32 (0), 17.91	2642 (8)
SMMALA	32 (0), 9.73	2130 (0)

E.3 Gaussian Copula Process Volatility model

Table 36: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	196 (25), 1.00	15 (0)
HMC	43 (15), 11.68	660 (3)
SMMALA	54 (24), 1.46	210 (0)

Table 37: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	35 (2), 1.34	15 (0)
HMC	32 (0), 8.17	2641 (8)
SMMALA	33 (0), 2.79	2130 (0)

Table 38: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	221 (21), 1.00	15 (0)
HMC	44 (17), 18.85	660 (3)
SMMALA	61 (37), 5.11	210 (0)

Table 39: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	44 (9), 1.22	15 (0)
HMC	32 (0), 15.64	2642 (8)
SMMALA	32 (0), 6.61	2130 (0)

E.4 Ordinal regression with GP priors

Table 40: $n = 100$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	242 (42), 1.00	15 (0)
HMC	44 (16), 11.04	660 (3)
SMMALA	45 (14), 3.17	210 (0)

Table 41: $n = 100$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	36 (2), 1.43	15 (0)
HMC	32 (0), 18.09	2641 (8)
SMMALA	32 (0), 9.45	2130 (0)

Table 42: $n = 400$, $d = 2$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	235 (20), 1.00	15 (0)
HMC	43 (16), 20.07	660 (3)
SMMALA	44 (13), 6.84	210 (0)

Table 43: $n = 400$, $d = 10$

Sampler	ESS (sd), $\hat{R}$	$10^3 \# O(n^3)$
MH	70 (15), 2.49	15 (0)
HMC	33 (0), 678.69	2642 (8)
SMMALA	32 (0), 76.67	2130 (0)

F Results - efficiency in sampling $\mathbf{f}, \boldsymbol{\theta}$ - Comparing ASIS with SA, AA, and KHR

F.1 Logistic regression

Table 44: $n = 100$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	779 (173)	153 (23)	1.00
ASIS ELL SS	329 (55)	115 (25)	1.01
SA ELL SS	161 (65)	34 (1)	1.75
AA ELL SS	304 (69)	99 (23)	1.01
KHR ELL SS	186 (35)	45 (4)	1.04

Table 45: $n = 100$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	134 (14)	52 (4)	1.03
ASIS ELL SS	75 (9)	46 (3)	1.04
SA ELL SS	71 (11)	36 (2)	1.30
AA ELL SS	79 (10)	45 (3)	1.04
KHR ELL SS	253 (43)	44 (3)	1.05

Table 46: $n = 400$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	142 (14)	59 (5)	1.02
ASIS ELL SS	81 (14)	42 (4)	1.03
SA ELL SS	78 (16)	33 (1)	3.49
AA ELL SS	83 (11)	43 (3)	1.08
KHR ELL SS	37 (2)	33 (2)	3.23

Table 47: $n = 400$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	120 (10)	37 (2)	1.42
ASIS ELL SS	53 (7)	36 (2)	1.30
SA ELL SS	64 (11)	33 (0)	2.41
AA ELL SS	58 (10)	38 (4)	1.30
KHR ELL SS	53 (3)	37 (1)	1.13

F.2 Log-Gaussian Cox model

Table 48: $n = 100$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	242 (56)	65 (9)	1.02
ASIS ELL SS	77 (6)	45 (2)	1.07
SA ELL SS	71 (5)	34 (2)	1.95
AA ELL SS	77 (7)	41 (2)	1.04
KHR ELL SS	40 (6)	35 (2)	1.85

Table 49: $n = 100$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	184 (37)	42 (5)	1.11
ASIS ELL SS	36 (1)	39 (3)	1.20
SA ELL SS	36 (1)	40 (4)	1.25
AA ELL SS	36 (1)	34 (1)	2.44
KHR ELL SS	33 (0)	48 (6)	3.45

Table 50: $n = 400$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	118 (30)	43 (5)	1.12
ASIS ELL SS	49 (5)	36 (2)	1.30
SA ELL SS	47 (4)	35 (3)	5.02
AA ELL SS	51 (5)	37 (2)	1.36
KHR ELL SS	35 (2)	36 (6)	2.39

Table 51: $n = 400$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	160 (14)	37 (2)	1.19
ASIS ELL SS	33 (0)	36 (2)	1.48
SA ELL SS	33 (0)	35 (2)	3.12
AA ELL SS	33 (0)	33 (1)	2.67
KHR ELL SS	32 (0)	41 (3)	11.31

F.3 Gaussian Copula Process Volatility model

Table 52: $n = 100$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	272 (38)	70 (9)	1.02
ASIS ELL SS	68 (6)	47 (3)	1.04
SA ELL SS	53 (8)	35 (1)	2.10
AA ELL SS	61 (5)	41 (4)	1.06
KHR ELL SS	36 (3)	36 (5)	1.57

Table 53: $n = 100$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	245 (60)	43 (4)	1.10
ASIS ELL SS	36 (1)	41 (5)	1.08
SA ELL SS	36 (1)	37 (1)	1.28
AA ELL SS	36 (2)	34 (1)	1.69
KHR ELL SS	34 (1)	37 (5)	5.70

Table 54: $n = 400$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	144 (22)	43 (5)	1.05
ASIS ELL SS	45 (3)	38 (2)	1.21
SA ELL SS	44 (4)	36 (2)	6.96
AA ELL SS	47 (1)	36 (2)	1.22
KHR ELL SS	34 (1)	34 (3)	5.24

Table 55: $n = 400$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	154 (46)	42 (8)	1.27
ASIS ELL SS	33 (0)	37 (3)	1.76
SA ELL SS	33 (0)	35 (3)	4.17
AA ELL SS	33 (0)	33 (1)	2.21
KHR ELL SS	32 (0)	41 (3)	10.47

F.4 Ordinal regression with GP priors

Table 56: $n = 100$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	143 (20)	49 (3)	1.07
ASIS ELL SS	53 (5)	40 (3)	1.10
SA ELL SS	47 (4)	35 (2)	3.93
AA ELL SS	51 (5)	38 (2)	1.35
KHR ELL SS	40 (2)	38 (4)	5.48

Table 57: $n = 100$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	704 (265)	47 (6)	2.39
ASIS ELL SS	37 (1)	45 (4)	1.09
SA ELL SS	36 (1)	43 (3)	1.12
AA ELL SS	36 (1)	33 (1)	2.22
KHR ELL SS	34 (1)	36 (2)	1.18

Table 58: $n = 400$, $d = 2$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	104 (10)	40 (3)	1.09
ASIS ELL SS	43 (3)	36 (4)	1.87
SA ELL SS	40 (3)	41 (5)	7.93
AA ELL SS	42 (3)	34 (1)	1.87
KHR ELL SS	35 (2)	38 (4)	14.67

Table 59: $n = 400$, $d = 10$

Sampler	ESS $\mathbf{f}$	ESS $\boldsymbol{\theta}$	$\hat{R}$
ASIS HMC v2	393 (190)	38 (4)	2.20
ASIS ELL SS	33 (0)	38 (2)	1.56
SA ELL SS	33 (0)	38 (5)	2.80
AA ELL SS	32 (0)	33 (0)	5.63
KHR ELL SS	32 (0)	42 (7)	6.37