Noname manuscript No.
(will be inserted by the editor)

A Comparative Evaluation of Stochastic-based Inference Methods for Gaussian Process Models

M. Filippone · M. Zhong · M. Girolami

Received: date / Accepted: date

Abstract Gaussian Process (GP) models are extensively used in data analysis given their flexible modeling capabilities and interpretability. The fully Bayesian treatment of GP models is analytically intractable, and therefore it is necessary to resort to either deterministic or stochastic approximations. This paper focuses on stochastic-based inference techniques. After discussing the challenges associated with the fully Bayesian treatment of GP models, a number of inference strategies based on Markov chain Monte Carlo methods are presented and rigorously assessed. In particular, strategies based on efficient parameterizations and efficient proposal mechanisms are extensively compared on simulated and real data on the basis of convergence speed, sampling efficiency, and computational cost.

Keywords Bayesian inference · Gaussian Processes · Markov chain Monte Carlo · hierarchical models · latent variable models

1 Introduction

Gaussian Process (GP) models represent a class of models that are popular in data analysis due to the associated flexibility and interpretability. Both these features are a direct consequence of their rich parameterization. Flexibility is due to the nonparametric prior over latent variables conditioning observations, whereas interpretability is due to the parameterization of the structure associated with the latent variables. Observations are conditionally independent given a set of jointly

Maurizio Filippone
School of Computing Science, University of Glasgow, United Kingdom.
E-mail: maurizio.filippone@glasgow.ac.uk

Mingjun Zhong
Department of Biomedical Engineering, Dalian University of Technology, P.R. China
E-mail: mingjun.zhong@gmail.com

Mark Girolami
Department of Statistical Science, University College London, United Kingdom.
E-mail: girolami@stats.ucl.ac.uk

Gaussian latent variables, and are assumed to be distributed according to the particular type of data being modeled. The covariance structure of the latent variables is then parameterized by a set of hyper-parameters that characterizes the covariance of the input vectors in terms of length-scales and intensity of interaction. GP models comprise a large set of models, and this paper focuses in particular on Logistic Regression with GP priors (`LRG`) (Rasmussen and Williams, 2006), Log-Gaussian Cox models (`LCX`) (Møller et al., 1998), Stochastic Volatility models with GP priors (`VLT`) (Wilson and Ghahramani, 2010), and Ordinal Regression with GP priors (`ORD`) (Chu and Ghahramani, 2005).

Exact inference in GP models is analytically intractable. Most of the work to tackle such intractability focuses on deterministic approximations to integrate out latent variables; those approaches include the Laplace Approximation (LA) (Tierney and Kadane, 1986), Expectation Propagation (EP) (Minka, 2001), and mean field approximations (Opper and Winther, 2000) (see, e.g., Rasmussen and Williams (2006) for an extensive presentation of such approximations, and Kuss and Rasmussen (2005) for their assessment on `LRG` models). Deterministic approximations provide a computationally tractable way to integrate out latent variables, but it is not possible to quantify the error that they introduce in the quantification of uncertainty in predictions (although EP for `LRG` is reported to be very accurate in Kuss and Rasmussen (2005)); also, those methods target the integration of latent variables only.

In the direction of providing a fully Bayesian treatment of GP models, it is necessary to integrate out latent variables as well as hyper-parameters, and this is usually done by quadrature methods (Cseke and Heskes, 2011; Rue et al., 2009), thus limiting the number of hyper-parameters that can be employed in GP models.

Based on those considerations, this paper focuses on non-deterministic methods to carry out inference in GP models, and in particular on stochastic based approximations based Markov Chain Monte Carlo (MCMC) methods. The use of MCMC based inference methods is appealing as it provides asymptotic guarantees of convergence to exact inference. In practice, this translates into the possibility of achieving results with the desired level of accuracy (Flegal et al., 2007). Unfortunately, the use of MCMC methods for inference in GP models is extremely difficult. The aim of this paper is to discuss the challenges associated with MCMC based inference for GP models, and compare a number of strategies that have been proposed in the literature to tackle them. A preliminary version of this work can be found in Filippone et al. (2012)[1].

To the best of our knowledge, this work (i) is the first attempt to extensively assess the state-of-the-art in stochastic-based inference methods for GP models, and (ii) sets the bar for new MCMC methods for inference in GP models. Along with those contributions, this paper presents (iii) a variant of the Hybrid Monte Carlo algorithm that outperforms state-of-the-art methods to sample from the posterior distribution of the latent variables, and (iv) tests the combination of parameterizations, as recently proposed in Yu and Meng (2011), in the case of GP models.

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

1.1 Gaussian Process Models

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of n input vectors described by a set of d covariates $\mathbf{x}_i \in \mathbb{R}^d$, associated with observed responses $\mathbf{y} = \{y_1, \ldots, y_n\}$. In GP models, the generative process modeling the observed data $\mathbf{y}$ given X is as follows. Observations are assumed to be conditionally independent given a set of n latent variables $\mathbf{f} = \{f_1, \ldots, f_n\}$, 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. This can be translated into a likelihood function of the form $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i)$, where for generality the distribution $p(y_i|f_i)$ is left unspecified.

In this work, latent variables are assumed to be drawn from a zero mean GP prior with covariance function k. The GP prior is a prior over functions, and the covariance structure given by k specifies the characteristics of such functions (i.e., degree of smoothness and marginal variance). Let k be parameterized by a vector of hyper-parameters $\boldsymbol{\theta} = (\sigma, \psi_{\tau_1}, \ldots, \psi_{\tau_d})$, and assume:

$$k(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}) = \sigma q(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\psi}_{\boldsymbol{\tau}}) = \sigma \exp\left[-\frac{1}{2}\sum_{r=1}^{d} \frac{(\mathbf{x}_i - \mathbf{x}_j)^2_{(r)}}{\exp(\psi_{\tau_r})^2}\right] \tag{1}$$

with $\exp(\psi_{\tau_r})$ defining the length-scale of the interaction between the input vectors for the rth covariate and σ giving the marginal variance for latent variables. This type of covariance can be used for Automatic Relevance Determination (ARD) (Mackay, 1994) of the covariates, as the values $\tau_i = \exp(\psi_{\tau_i})$ 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. Exponentiation of the hyper-parameters is convenient, so that standard MCMC transition operators can be employed for ψ_{τ_i} thus avoiding dealing with boundary conditions or non-standard MCMC proposals (Robert and Casella, 2005). 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 model is fully specified by choosing a prior $p(\boldsymbol{\theta})$ for the hyper-parameters. The model structure is therefore hierarchical, with hyper-parameters conditioning the latent variables that, in turn, condition observations, so that $p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

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

$$p(y_*|\mathbf{y}) = \int\int\int p(y_*|f_*)p(f_*|\mathbf{f}, \boldsymbol{\theta})p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{y})df_* d\mathbf{f} d\boldsymbol{\theta} \tag{2}$$

The left hand side of Eq. 2 is a full probability distribution characterizing the uncertainty in predicting y_* given the GP modeling assumption.

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 obtain a Monte Carlo estimate of the predictive distribution as follows:

$$p(y_*|\mathbf{y}) \simeq \frac{1}{N}\sum_{i=1}^{N} \int p(y_*|f_*)p(f_*|\mathbf{f}^{(i)}, \boldsymbol{\theta}^{(i)})df_* \tag{3}$$

where N denotes the number of samples used to compute the estimate. In Eq. 3 we denoted the ith samples from the posterior distribution of $\mathbf{f}$ and $\boldsymbol{\theta}$ obtained by means of MCMC methods by $\mathbf{f}^{(i)}$ and $\boldsymbol{\theta}^{(i)}$. Note that the remaining integral is univariate and it is generally easy to evaluate.

1.2 Challenges in MCMC based inference for GP models

Sampling from the posterior of latent variables and hyper-parameters by joint proposals is not feasible; it is extremely unlikely to propose a set of latent variables and hyper-parameters that are compatible with each other and observed data. This forces one to consider schemes such as Gibbs sampling, where groups of variables are updated one at time, leading to the following challenges:

(i) Due to the hierarchical structure of GP models, chains converge slowly and mix poorly if the coupling effect between the groups of variables is not dealt with properly. This requires some form of reparameterization or clever proposal mechanism that efficiently decouples the dependencies between the groups of variables. This effect has drawn a lot of attention in the case of hierarchical models in general (Yu and Meng, 2011), and recently in GP models 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) a parameterization based on auxiliary data is proposed that aims at reducing the coupling between the two groups of variables. Other ideas involve the use of reparameterizations based on whitening the latent variables; in the terminology of Yu and Meng (2011), this corresponds to employing the so called Ancillary Augmentation (AA) parameterization. Recently, Yu and Meng (2011) proposed to interweave parameterizations characterized by complementary features in order to boost sampling efficiency. Parameterizations can be complementary in the sense that they offer better performance in either strong or weak data limits; the idea of combining parameterizations is to achieve high sampling efficiency in both strong and weak data scenarios. We are interested in comparing the methods in Knorr-Held and Rue (2002); Murray and Adams (2010) and Yu and Meng (2011) applied to GP models. Another possibility would be to approximately integrate out latent variables and obtain samples from the corresponding approximate posterior of hyper-parameters. For GP classification this might be a sensible thing to do, as the Expectation Propagation approximation has been reported to be very accurate Kuss and Rasmussen (2005); however, this is peculiar to GP classification and for general GP models it may not be the case.

(ii) Sampling hyper-parameters and latent variables cannot be done using exact Gibbs steps, and it requires proposals that are accepted/rejected based on a Hastings ratio, leading to a waste of expensive computations. Transition operators characterized by acceptance mechanisms embedded in a Gibbs sampler, are usually referred to as Metropolis-within-Gibbs operators. Designing proposals that guarantee high acceptance and independence between samples is extremely challenging, especially because latent variables can have dimensions in the order of hundreds or thousands. We will compare several transition operators, for different steps of the Gibbs sampler, with the aim of gaining insights about ways to strike a good balance between efficiency and computational cost. We will consider transition operators characterized by proposal mechanisms with increasing complexity, and in particular the Metropolis-Hastings (MH) operator which is based on random walk types of proposals, Hybrid Monte Carlo (HMC) which uses the gradient of the log-density of interest, and manifold methods (Girolami and Calderhead, 2011) which use curvature information (i.e., second derivatives of the log-density).

The paper is organized as follows: Sections 2 and 3 report the parameterization strategies and the transition operators considered in this work. Sections 4 and 5 report an extensive comparison of those strategies and transition operators, on simulated and real data, on the basis of efficiency, convergence speed and computational complexity; section 6 concludes the paper. For the sake of readability, most of the technical derivations can be found in the appendices.

2 Dealing with the hierarchical structure of GP models

2.1 Sufficient and Ancillary Augmentation

From a generative perspective, the model structure is hierarchical with latent variables representing sufficient statistics for the hyper-parameters. This parameterization is referred to as Sufficient Augmentation (SA) in Yu and Meng (2011) and allows one to express the joint density as

$$\text{SA} \qquad p(\mathbf{y}, \mathbf{f}, \boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \tag{4}$$

It is also possible to introduce the decomposition of the matrix Q into the product of two factors LL^{T}, and view the generation of the latent variables as $\mathbf{f} = \sqrt{\sigma} L \boldsymbol{\nu}$ with $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{\nu}|\mathbf{0}, I)$, which implies that $\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\mathbf{0}, K)$. 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. In this way, $\boldsymbol{\nu}$ is ancillary for $\boldsymbol{\theta}$ and it is possible to express the joint density as

$$\text{AA} \qquad p(\mathbf{y}, \boldsymbol{\nu}, \boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\nu}, \boldsymbol{\theta}) p(\boldsymbol{\nu}) p(\boldsymbol{\theta}) \tag{5}$$

This parameterization is called Ancillary Augmentation (AA) in the terminology of Yu and Meng (2011). In Murray and Adams (2010) SA and AA are referred to as unwhitened and whitened parameterizations respectively. Weak and strong data limits can influence the efficiency in sampling using either parameterization. For this reason, it is important to choose an efficient parameterizations for the particular problem under study and for the available amount of data, as both these aspects can dramatically influence efficiency and convergence speed of the chains.

2.2 Ancillarity-Sufficiency Interweaving Strategy - ASIS

In this section we briefly review the main results presented in Yu and Meng (2011) on the combination of parameterizations to improve convergence and efficiency of MCMC methods, and we will illustrate how these results can be applied to GP models. Intuitively, combining parameterizations seems promising to take the best from them in both weak and strong data limits, or at least, to avoid the possibility that chains do not converge because of the wrong choice of parameterization. Alternating the sampling in the SA and AA parameterizations is the most obvious way of combining the two parameterizations, but as recently investigated in Yu and Meng (2011), interweaving SA and AA is actually a more promising way forward. From a theoretical perspective, the geometric rate of convergence r of the scheme when the parameterizations 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 $r \leq \max(r_1, r_2)$, combining the two parameterizations 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 two parameterizations, so it is possible to make the combined scheme converge quickly even if neither of the individual schemes do. In general, this result is quite remarkable, as once different reparameterizations are available, combining them using the interweaving strategy is simple to implement, and can dramatically boost sampling efficiency. In GP models, the ASIS scheme amounts to interweaving SA and AA updates, that following Yu and Meng (2011) yields:

$$\mathbf{f}|\mathbf{y},\boldsymbol{\theta} \quad \longrightarrow \quad \boldsymbol{\theta}|\mathbf{f} \quad \longrightarrow \quad \boldsymbol{\nu} = \sigma^{-1/2}L^{-1}\mathbf{f} \quad \longrightarrow \quad \boldsymbol{\theta}|\mathbf{y},\boldsymbol{\nu} \tag{6}$$

2.3 Knorr-Held and Rue (KHR)

The idea underpinning KHR, is to jointly sample parameters and latent variables as follows. Firstly, a set of hyper-parameters $\boldsymbol{\theta}'|\boldsymbol{\theta}$ is proposed and secondly a set of latent variables conditioned on the new set of hyper-parameters, namely $\mathbf{f}'|\mathbf{y},\boldsymbol{\theta}'$, is proposed. The proposal $(\boldsymbol{\theta}',\mathbf{f}')$ is then jointly accepted or rejected according to a standard Hastings ratio. The key idea is to avoid making the proposal $\boldsymbol{\theta}'$ accepted on the basis of $\mathbf{f}$ to avoid the strong coupling effect due to the hierarchical nature of the model. KHR was proposed in applications making use of Gaussian Markov Random Fields, and we will discuss the application of this idea for GP models in the section reporting the experiments. In order to avoid difficulties in devising a proposal for sampling from $\mathbf{f}'|\mathbf{y},\boldsymbol{\theta}')$, here we set the proposal as the Gaussian obtained by constructing a Laplace approximation to $p(\mathbf{f}|\mathbf{y},\boldsymbol{\theta}')$.

2.4 Surrogate Method (SURR)

In the SURR method (Murray and Adams, 2010), a set of auxiliary latent variables $\mathbf{g}$ is introduced as a noisy version of $\mathbf{f}$; in particular, $p(\mathbf{g}|\mathbf{f},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{g}|\mathbf{f},S_{\boldsymbol{\theta}})$. This construction yields a conditional distribution for $\mathbf{f}$ of the form $p(\mathbf{f}|\mathbf{g},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}|\mathbf{m},R)$, with $R = S_{\boldsymbol{\theta}} - S_{\boldsymbol{\theta}}(S_{\boldsymbol{\theta}} + K)^{-1}S_{\boldsymbol{\theta}}$ and $\mathbf{m} = RS_{\boldsymbol{\theta}}^{-1}\mathbf{g}$. After decomposing $R = DD^{\mathrm{T}}$, the sampling of $\boldsymbol{\theta}$ is then conditioned on the variables $\boldsymbol{\eta}$ defined as $\mathbf{f} = D\boldsymbol{\eta} + \mathbf{m}$. The covariance $S_{\boldsymbol{\theta}}$ is constructed to be diagonal with elements obtained by matching the posterior for each latent variable individually or by Taylor approximations (see Murray and Adams (2010) for details).

3 MCMC transition operators considered in this work

This section presents the transition operators considered in this work. We are interested in understanding whether and to what extent employing proposal mechanisms making use of gradient or curvature information of the target density improves sampling efficiency and speed of convergence with respect to computational complexity. We therefore consider transition operators with increasing complexity, and in particular the Metropolis-Hastings (MH) operator which is based on random walk types of proposals, the Hybrid Monte Carlo (HMC) operator which uses gradient information, and the Simplified Manifold Metropolis Adjusted Langevin Algorithm (SMMALA) operator which is one of the simplest manifold MCMC methods proposed in Girolami and Calderhead (2011) using curvature information.

For the sake of clarity, we will focus on the transitions operators for $\mathbf{f}$, but the same operators can be easily applied to $\boldsymbol{\theta}$. We will first present MH, HMC, and SMMALA, and we will then discuss Elliptical Slice Sampling and a few variants of MH and HMC that have been specifically proposed for sampling $\mathbf{f}$, and do not have counterparts for $\boldsymbol{\theta}$. In the case of latent variables, the operators aim to leave the posterior $p(\mathbf{f}|\mathbf{y},\boldsymbol{\theta})$ invariant; in the remainder of this work, $W(\mathbf{f})$ is defined as $\log[p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\theta})]$, which equals the log of the desired target density up to constants. In the case of hyper-parameters we can define the invariant distribution according to the chosen parameterization and apply the operators presented here for sampling $\boldsymbol{\theta}$ rather than $\mathbf{f}$.

3.1 Metropolis-Hastings - MH

The Metropolis-Hastings transition operator employs a proposal mechanism $g(\mathbf{f}'|\mathbf{f})$ based on a random walk (Robert and Casella, 2005). A common choice is to use a multivariate Gaussian proposal with covariance Σ centered at the former position $\mathbf{f}$, thus taking the form $g(\mathbf{f}'|\mathbf{f}) = \mathcal{N}(\mathbf{f}'|\mathbf{f},\Sigma)$. For such a symmetric proposal mechanism, $\mathbf{f}'$ is then accepted with probability $\min\left\{1, \exp(W(\mathbf{f}') - W(\mathbf{f}))\right\}$.

3.2 Hybrid Monte Carlo - HMC

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

$$H(\mathbf{f},\mathbf{p}) = -W(\mathbf{f}) + \frac{1}{2}\log(|M|) + \frac{1}{2}\mathbf{p}^{\mathrm{T}}M^{-1}\mathbf{p} + \text{const.} \tag{7}$$

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

$$\frac{d\mathbf{f}}{dt} = \frac{\partial H}{\partial \mathbf{p}} = M^{-1}\mathbf{p} \qquad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{f}} = \nabla_{\mathbf{f}} W \tag{8}$$

Given that there is no friction, the energy will be conserved during the motion of the particle. Solving Hamilton's equations directly for general potential fields, however, is analytically intractable, and therefore it is necessary to resort to schemes where time is discretized. The leapfrog integrator discretizes the dynamics in λ steps, also known as leapfrog steps, and is volume preserving and reversible (see Neal (1993) for details). The leapfrog integrator yields an update of $(\mathbf{f}, \mathbf{p})$ into $(\mathbf{f}_{(\lambda)}, \mathbf{p}_{(\lambda)})$. The discretization introduces an approximation such that the total energy is not conserved, so a Metropolis accept/reject step of the form $\min\{1, \exp(-H(\mathbf{f}_{(\lambda)}, \mathbf{p}_{(\lambda)}) + H(\mathbf{f}, \mathbf{p}))\}$ is needed to ensure that HMC samples from the correct invariant distribution. The HMC transition operator is reported in Algorithm 1.

Algorithm 1 HMC transition operator when $M = L_M L_M^{\mathrm{T}}$

```
$\mathbf{f}_{(0)} = \mathbf{f}$; $\mathbf{p}_{(0)} \sim \mathcal{N}(\mathbf{p}_{(0)}|\mathbf{0}, M)$        ▷ $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$; $\mathbf{p}_{(0)} = L_M \mathbf{z}$
$\lambda = \mathrm{sample}[1, \ldots, \lambda_{\max}]$
for $(t = 0 \text{ to } \lambda - 1)$ do
    $\mathbf{p}_{(t+1/2)} = \mathbf{p}_{(t)} + \frac{\varepsilon}{2}\nabla_{\mathbf{f}} W(\mathbf{f}_{(t)})$
    $\mathbf{f}_{(t+1)} = \mathbf{f}_{(t)} + \varepsilon M^{-1}\mathbf{p}_{(t+1/2)}$        ▷ $M^{-1}\mathbf{p} = \mathtt{bcksub}(L_M^{\mathrm{T}}, (\mathtt{fwdsub}(L_M, \mathbf{p})))$
    $\mathbf{p}_{(t+1)} = \mathbf{p}_{(1/2)} + \frac{\varepsilon}{2}\nabla_{\mathbf{f}} W(\mathbf{f}_{(t+1)})$
end for
$r = \min\left\{0, H(\mathbf{f}_{(0)}, \mathbf{p}_{(0)}) - H(\mathbf{f}_{(\lambda)}, \mathbf{p}_{(\lambda)})\right\}$        ▷ $\log|M| = 2\sum_i \log(L_M)_{ii}$
                                                                                   ▷ $\mathbf{p}^{\mathrm{T}} M^{-1}\mathbf{p} = \|\mathtt{fwdsub}(L_M, \mathbf{p})\|^2$
$u \sim \mathrm{Exp}(u|1)$
if $(r > -u)$ then return $\mathbf{f}_{(\lambda)}$
else return $\mathbf{f}_{(0)}$
```

3.3 Manifold MCMC - Simplified Manifold MALA - SMMALA

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 (FI) matrix. The FI matrix and the Christoffel symbols are the key quantities in information geometry as they characterize the curvature and the connection on the statistical 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:

$$G(\boldsymbol{\psi}) = \mathrm{E}_{p(\mathbf{y}|\boldsymbol{\psi})}\left[\left(\nabla_{\boldsymbol{\psi}}\mathcal{L}\right)\left(\nabla_{\boldsymbol{\psi}}\mathcal{L}\right)^{\mathrm{T}}\right] = -\mathrm{E}_{p(\mathbf{y}|\boldsymbol{\psi})}[\nabla_{\boldsymbol{\psi}}\nabla_{\boldsymbol{\psi}}\mathcal{L}] \tag{9}$$

By definition, the FI matrix is positive semidefinite, and can be considered as the natural metric on S.

In the case of GP models that are hierarchical we need to consider the statistical manifolds associated with the two levels of the hierarchy separately. Let's focus on the statistical manifold associated with the model for $\mathbf{y}$ given $\mathbf{f}$. The manifold MALA (MMALA) algorithm (Girolami and Calderhead, 2011) defines a Langevin diffusion with stationary distribution $p(\mathbf{f}|\boldsymbol{\theta}, \mathbf{y})$ on the Riemann manifold of density functions, characterized by a metric tensor denoted as $G_{\mathbf{f},\mathbf{f}}$. By employing a first order Euler integrator to solve the diffusion, a proposal mechanism with density $g(\mathbf{f}'|\mathbf{f}) = \mathcal{N}(\mathbf{f}'|\boldsymbol{\mu}(\mathbf{f}, \epsilon), \epsilon^2 G_{\mathbf{f},\mathbf{f}}^{-1})$ is obtained, where ϵ is the integration step size, a parameter which needs to be tuned, and the dth component of the mean function $\boldsymbol{\mu}(\mathbf{f}, \epsilon)_d$ is

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

where $\Gamma_{i,j}^{d}$ are the Christoffel symbols of the metric in local coordinates (Amari and Nagaoka, 2000). 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 a standard Metropolis accept/reject step is employed to correct this bias.

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 will not consider RM-HMC or MMALA, as they both require the derivatives of the FI matrix 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

$$\boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon) = \mathbf{f} + \frac{\epsilon^2}{2} G_{\mathbf{f},\mathbf{f}}^{-1}\nabla_{\mathbf{f}} W(\mathbf{f}) \tag{10}$$

Furthermore, in the last subsection of this section we will present two variants of HMC that bear some similarities with RM-HMC but are computationally cheaper. The SMMALA transition operator is sketched in Algorithm 2.

Algorithm 2 SMMALA transition operator

```
$\boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon) = \mathbf{f} + \frac{\epsilon^2}{2} G_{\mathbf{f},\mathbf{f}}^{-1}\nabla_{\mathbf{f}} W(\mathbf{f})$    ▷ $G_{\mathbf{f},\mathbf{f}} = L_G L_G^{\mathrm{T}}$
    ▷ $G_{\mathbf{f},\mathbf{f}}^{-1}\nabla_{\mathbf{f}} W(\mathbf{f}) = \texttt{bcksub}(L_G^{\mathrm{T}}, (\texttt{fwdsub}(L_G, \nabla_{\mathbf{f}} W(\mathbf{f}))))$
$\mathbf{f}' \sim \mathcal{N}(\mathbf{f}'|\boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon), \epsilon^2 G_{\mathbf{f},\mathbf{f}}^{-1})$    ▷ $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$; $\mathbf{f}' = \varepsilon\, \texttt{bcksub}(L_G^{\mathrm{T}}, \mathbf{z}) + \boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon)$
$r = \min\{0, W(\mathbf{f}') - W(\mathbf{f}) + \log[g(\mathbf{f}|\mathbf{f}')] - \log[g(\mathbf{f}'|\mathbf{f})]\}$    ▷ $\log|G_{\mathbf{f},\mathbf{f}}| = 2\sum_i \log(L_G)_{ii}$
    ▷ $(\mathbf{f}' - \boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon))^{\mathrm{T}} G_{\mathbf{f},\mathbf{f}}^{-1}(\mathbf{f}' - \boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon)) = \|\texttt{fwdsub}(L_G, (\mathbf{f}' - \boldsymbol{\mu}_{\mathrm{s}}(\mathbf{f}, \epsilon)))\|^2$
$u \sim \mathrm{Exp}(u|1)$
if ($r > -u$) then return $\mathbf{f}'$
else return $\mathbf{f}$
```

3.4 Elliptical Slice sampling - ELL-SS

Elliptical Slice Sampling (ELL-SS) has been proposed in Murray et al. (2010) to draw samples for $\mathbf{f}$ in GP models, and 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, when constrained on an ellipse. For completeness, we report the transition operator in Algorithm 3 and we refer the reader to Murray et al. (2010) for further details. Note that ELL-SS is quite appealing as it returns a sample which does not need to be accepted or rejected (in fact, a rejection mechanism is implicit within step 5), and the proposal mechanism does not have any free parameters that need tuning.

Algorithm 3 ELL-SS transition operator

```
 1: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, K)$
 2: $u \sim \mathrm{Exp}(u|1)$        $\eta = \log p(\mathbf{y}|\mathbf{f}) - u$        ▷ Set a threshold on the log-likelihood
 3: $\alpha \sim U[0, 2\pi]$        $[\alpha_{\min}, \alpha_{\max}] = [\alpha - 2\pi, \alpha]$        ▷ Define the bracket
 4: $\mathbf{f}' = \mathbf{f}\cos(\alpha) + \mathbf{z}\sin(\alpha)$
 5: if ($\log p(\mathbf{y}|\mathbf{f}') > \eta$) then return $\mathbf{f}'$
 6: else        ▷ Shrink the bracket
 7:     if ($\alpha < 0$) then $\alpha_{\min} = 0$
 8:     else $\alpha_{\max} = 0$
 9:     $\alpha \sim U[\alpha_{\min}, \alpha_{\max}]$
10:     Go to 4
```

3.5 Scaled versions of MH - MH v1 and MH v2

Due to the strong correlation of latent variables imposed by the GP prior, employing a MH operator with an isotropic covariance to sample latent variables leads to extremely poor efficiency. In order to overcome this problem, Neal (1999) proposed two versions of MH that we will denote by MH v1 and MH v2. In MH v1, a set of latent variables $\mathbf{z}$ is drawn from the GP prior $\mathbf{z} \sim \mathcal{N}(\mathbf{z}|0, K)$, and the proposal is constructed as follows:

$$\mathbf{f}' = \mathbf{f} + \alpha\,\mathbf{z} \tag{11}$$

where the parameter α controls the degree of update. In MH v2, instead, the proposal is as follows:

$$\mathbf{f}' = \sqrt{1 - \alpha^2}\,\mathbf{f} + \alpha\,\mathbf{z} \tag{12}$$

In the latter case, given that the proposal satisfies detailed balance with respect to the prior, the acceptance has to be based on the likelihood alone.

3.6 Scaled versions of HMC - HMC v1 and HMC v2

By a similar argument as in MH, it is possible to introduce scaled versions of HMC that reduce the correlation between latent variables. This can be done by setting the mass matrix of HMC according to the precision of the posterior distribution of latent variables. Similarly, from an information geometric perspective, it is sensible to whiten latent variables according to the metric tensor of the statistical manifold. We notice that the metric tensor associated to the model for $\mathbf{y}$ given $\mathbf{f}$ is K^{-1} plus a diagonal matrix which is a function of $\mathbf{f}$ (see appendix A for full details). Whitening with respect to that metric tensor would be computationally very expensive for GP models, as it would require the simulation of the Hamiltonian dynamics on a manifold with a position-dependent curvature; this is implemented by RM-HMC which requires the derivatives of the metric tensor as well as implicit leapfrog iterations (Girolami and Calderhead, 2011). In order to reduce the computational cost, we propose the following two options: (i) to approximate the diagonal term to be independent of $\mathbf{f}$ so that $M^{-1} = (K^{-1} + C)^{-1} = C^{-1} - C^{-1}(K + C^{-1})^{-1}C^{-1}$ with C diagonal and independent of $\mathbf{f}$; we call this variant HMC v1. (ii) to ignore the diagonal part of the metric tensor and set $M^{-1} = K$; we call this variant HMC v2. In HMC v1, one simple way to make C independent of $\mathbf{f}$ is to compute it for the GP prior mean (which is zero), as proposed, e.g., in Christensen et al. (2005); Vanhatalo and Vehtari (2007).

In both cases, it is possible to employ a standard and computationally efficient HMC proposal that captures part of the curvature of the statistical manifold. This is achieved by introducing a variant of HMC that, rather than using the Cholesky decomposition of the mass matrix, uses the decomposition of its inverse. We report this variant of the HMC transition operator in Algorithm 4.

In HMC v1, employing this formulation of HMC is convenient as computing the inverse of M is more stable than computing $M = K^{-1} + C$, that requires a potentially unstable inversion of K. HMC v1 requires the computation of the inverse of the mass matrix and its factorization each time a new value of $\boldsymbol{\theta}$ is proposed. In HMC v2, instead, no extra operations in $O(n^3)$ are required given that K is already factorized, thus making it computationally very convenient.

Algorithm 4 HMC transition operator when $M^{-1} = L_{M^{-1}} L_{M^{-1}}^{\mathrm{T}}$

```
$\mathbf{f}_{(0)} = \mathbf{f}$; $\mathbf{p}_{(0)} \sim \mathcal{N}(\mathbf{p}_{(0)}|\mathbf{0}, M)$    ▷ $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$; $\mathbf{p}_{(0)} = \mathtt{bcksub}(L_{M^{-1}}^{\mathrm{T}}, \mathbf{z})$
$\lambda = \mathrm{sample}[1, \ldots, \lambda_{\max}]$
for ($t = 0$ to $\lambda - 1$) do
    $\mathbf{p}_{(t+1/2)} = \mathbf{p}_{(t)} + \frac{\varepsilon}{2} \nabla_{\mathbf{f}} W(\mathbf{f}_{(t)})$
    $\mathbf{f}_{(t+1)} = \mathbf{f}_{(t)} + \varepsilon M^{-1} \mathbf{p}_{(t+1/2)}$    ▷ $M^{-1}\mathbf{p} = L_{M^{-1}}(L_{M^{-1}}^{\mathrm{T}} \mathbf{p})$
    $\mathbf{p}_{(t+1)} = \mathbf{p}_{(1/2)} + \frac{\varepsilon}{2} \nabla_{\mathbf{f}} W(\mathbf{f}_{(t+1)})$
end for
$r = \min\left\{0, H(\mathbf{f}_{(0)}, \mathbf{p}_{(0)}) - H(\mathbf{f}_{(\lambda)}, \mathbf{p}_{(\lambda)})\right\}$    ▷ $\log|M| = -2\sum_i \log(L_{M^{-1}})_{ii}$
                                                ▷ $\mathbf{p}^{\mathrm{T}} M^{-1} \mathbf{p} = \|L_{M^{-1}}^{\mathrm{T}} \mathbf{p}\|^2$
$u \sim \mathrm{Exp}(u|1)$
if ($r > -u$) then return $\mathbf{f}_{(\lambda)}$
else return $\mathbf{f}_{(0)}$
```

4 Results on simulated data

In this section, we first report a study on the efficiency and speed of convergence of different transition operators in sampling from posterior distribution of individual groups of variables in the SA and AA parameterization. Secondly, we report the same analysis to compare different parameterizations to obtain samples from the joint posterior distribution of $\mathbf{f}$ and $\boldsymbol{\theta}$.

4.1 Experimental setup

We simulated data from the four GP models considered in this work, namely: `LRG`, `LCX`, `VLT`, and `ORD`. We generated 10 data sets simulating from each of the four models for all combinations of $n = 100, 400$, and $d = 2, 10$, for a total of 160 distinct data sets. In order to isolate the effect of different likelihood functions in the results, we seeded the generation of the input data matrix X, hyper-parameters, and latent variables so that these were the same across different models. Covariates were generated uniformly in the unit hyper-cube, and the parameters used to generate latent variables were $\sigma = \exp(2)$, $\psi_{\tau_i} \sim U[-3, -1]$. We imposed Gamma priors on the length-scale parameters with shape a and rate b, $p(\tau_i) = \mathrm{Gam}(\tau_i | a = 1, b = 1)$. We imposed an inverse Gamma prior $p(\sigma) = \mathrm{invGam}(\sigma | a = 1, b = 1)$, where a and b are shape and scale parameters respectively on σ to exploit conjugacy in the SA parameterization.

In all the experiments we collected 20000 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). The transition operators for $\mathbf{f}$ had the following tuning parameters: α for MH v1 and MH v2, and ε for SMMALA and the variants of HMC which used a maximum of 10 leapfrog steps. The transition operators for $\boldsymbol{\theta}$ employed the following proposals: MH used a covariance $\Sigma = \alpha I$, HMC used a mass matrix $M = \alpha I$ and 10 maximum leapfrog steps, and SMMALA used a step-size ε. Convergence analysis was performed using the $\hat{R}$ potential scale reduction factor (Gelman and Rubin, 1992), which is a classic score used to assess convergence of MCMC algorithms. The computation of the $\hat{R}$ value is based on the within and between chain variances; a value close to one indicates that convergence is reached. The $\hat{R}$ value was computed based on 10 chains initialized from the prior to study what efficiency can be achieved without running preliminary simulations; this is different from the initialization procedure suggested in Gelman and Rubin (1992) that requires locating the modes of the target density. Due to the fairly diffuse priors on the length-scale parameters, we noticed difficulty in achieving convergence in some cases; we therefore initialized ψ_{τ_i} randomly in the interval $[-3, -1]$. The value of $\hat{R}$ was checked at 1000, 2000, 5000, 10000, 20000 iterations. We use the following procedure to compactly visualize the speed of convergence; we threshold the median value of $\hat{R}$ across 10 data sets at each checkpoint and use the following visual coding to report speed of convergence: ▯ $< 1.1 <$ ▂ $< 1.3 <$ ▅ $< 2 <$ ▮, so that ▯ indicates that $\hat{R} < 1.1$, ▂ indicates that $1.1 < \hat{R} < 1.3$, and so on. We then stack the rectangles associated to each checkpoint where we computed the value of $\hat{R}$, thus producing a sort of histogram of the median of $\hat{R}$ over the iterations. Efficiency of MCMC methods is compared based on the minimum of the Effective Sample Size (ESS) (Robert and Casella, 2005) computed across all the sampled variables. We then report its mean and standard deviation across the 10 chains and the 10 different data sets for each combination of size of the data set, dimensionality, and type of likelihood.

Table 1 Breakdown of the number of operations in $O(n^3)$ required to apply the transition operators considered in this work. #M, #I and #C represent number of multiplication of $n \times n$ matrices, inversions of $n \times n$ matrices, and number of Cholesky decompositions respectively. Counts are reported as functions of the number of iterations T and number of covariates d. In HMC, $\bar{\lambda}$ denotes the average number of leapfrog steps in one iteration.

	$\mathbf{f}\|\boldsymbol{\theta}, \mathbf{y}$			$\boldsymbol{\theta}\|\mathbf{f}$			$\boldsymbol{\theta}\|\boldsymbol{\nu}, \mathbf{y}$		
	#M	#I	#C	#M	#I	#C	#M	#I	#C
MH	0	0	1	0	0	T	0	0	T
HMC	0	0	1	0	$T\bar{\lambda}$	T	0	0	$T + Td\bar{\lambda}$
SMMALA	0	1	T	Td	T	T	0	0	$T + Td$
ELL-SS	0	0	1	–	–	–	–	–	–
MH v1	0	0	1	–	–	–	–	–	–
MH v2	0	0	1	–	–	–	–	–	–
HMC v1	0	1	2	–	–	–	–	–	–
HMC v2	0	0	1	–	–	–	–	–	–

We are also interested in statistically assessing which methods achieve faster convergence. In order to do so, we perform pairwise Mann-Whitney tests with significance level of 0.05 comparing the value of $\hat{R}$ at the last checkpoint for all the chains across 10 data sets. This allows us to obtain an ordering of methods in terms of convergence speed. In each table we include a row at the bottom reporting the result of such a test. We denote by $1|2$ situations where the method in row 1 of the corresponding table converges significantly faster than the method in row 2. Instead, the notation $1, 2$ is used when the method in row 1 does not converge significantly faster than the method in row 2.

As a measure of complexity, we counted the number of operations with complexity in $O(n^3)$, namely number of Cholesky factorizations of $n \times n$ matrices (#C), number of inversions of $n \times n$ matrices (#I)[2], and number of multiplications of $n \times n$ matrices (#M). We believe that this is a more reliable measure of complexity with respect to running time, as running time can be affected by several implementation details and other factors that are not directly related to the actual complexity of the algorithms.

4.2 Assessing the efficiency of samplers for individual groups of variables

In this section, we present an assessment of the efficiency of different transition operators for each group of variables using both SA and AA parameterizations. Computational complexity for all the operators considered in the next sections is summarized in Tab. 1, where T represents the number of iterations, d the number of covariates and $\bar{\lambda}$ the average number of leapfrog steps in HMC transition operators. In the following sub-sections we present results about the sampling of the latent variables and hyper-parameters separately.

Table 2 Comparison of transition operators to sample $\mathbf{f}|\mathbf{y}, \boldsymbol{\theta}$ for data generated from models with four different likelihoods. Minimum ESS is averaged over 10 chains for 10 different data sets for each value of n and d. The last row in each sub-table reports the result of the statistical test to assess which operators achieve significantly faster convergence.

LRG

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH v1	67 (15)	▥	47 (3)	▥	22 (7)	▥	8 (1)	▥
MH v2	204 (35)	▥	151 (7)	▥	67 (17)	▥	30 (2)	▥
SMMALA	756 (284)	▥	262 (30)	▥	457 (212)	▥	48 (5)	▥
ELL-SS	321 (61)	▥	241 (11)	▥	104 (25)	▥	50 (2)	▥
HMC v1	3395 (400)	▥	5163 (268)	▥	1352 (380)	▥	2962 (155)	▥
HMC v2	4004 (577)	▥	5225 (224)	▥	1566 (342)	▥	2995 (129)	▥
	6|5|3|4|2|1		6|5|3, 4|2|1		6|5|3|4|2|1		6|5|4|3|2|1	

LCX

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH v1	18 (16)	▥	6 (2)	▥	6 (5)	▥	1 (0)	▥
MH v2	23 (24)	▥	6 (2)	▥	8 (7)	▥	1 (0)	▥
SMMALA	217 (155)	▥	39 (4)	▥	258 (177)	▥	7 (1)	▥
ELL-SS	39 (42)	▥	11 (4)	▥	11 (11)	▥	2 (0)	▥
HMC v1	372 (277)	▥	188 (123)	▥	199 (200)	▥	81 (30)	▥
HMC v2	254 (197)	▥	188 (125)	▥	64 (37)	▥	80 (30)	▥
	6|5|3|4|2|1		6|5|3|4|1, 2		5|3, 6|4|2|1		6|5|3|4|2|1	

VLT

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH v1	28 (13)	▥	10 (2)	▥	15 (8)	▥	2 (0)	▥
MH v2	31 (16)	▥	12 (2)	▥	16 (8)	▥	2 (0)	▥
SMMALA	424 (216)	▥	117 (13)	▥	418 (127)	▥	61 (7)	▥
ELL-SS	46 (20)	▥	18 (4)	▥	22 (10)	▥	4 (1)	▥
HMC v1	1494 (667)	▥	449 (42)	▥	1384 (392)	▥	249 (25)	▥
HMC v2	418 (68)	▥	443 (39)	▥	183 (31)	▥	245 (25)	▥
	4|3|2|1, 5, 6		3|4|2|1, 5, 6		3, 4|2|1, 5, 6		3|6|4, 5|2|1	

ORD

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH v1	14 (8)	▥	6 (2)	▥	7 (5)	▥	1 (0)	▥
MH v2	14 (9)	▥	7 (2)	▥	7 (5)	▥	2 (0)	▥
SMMALA	48 (89)	▥	2 (0)	▥	107 (156)	▥	1 (0)	▥
ELL-SS	21 (11)	▥	10 (2)	▥	9 (5)	▥	2 (0)	▥
HMC v1	539 (650)	▥	472 (39)	▥	176 (200)	▥	257 (23)	▥
HMC v2	175 (54)	▥	483 (37)	▥	61 (22)	▥	255 (24)	▥
	6|5|3|4|1, 2		6|5|4|2|1|3		6|5|3|4|1, 2		6|5|2, 4|1, 3	

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

In this section we focus on the sampling from the posterior distribution of the latent variables $\mathbf{f}$. The results can be found in Tab. 2, and they were obtained

[2] This is a shorthand notation to denote a back and forward substitution of the identity matrix using Cholesky factors.

Table 3 Comparison of transition operators to sample $\boldsymbol{\theta}|\mathbf{f}$. Minimum ESS is averaged over 10 chains for 10 different data sets for each value of n and d. The last row reports the result of the statistical test to assess which operators achieve significantly faster convergence.

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH	2024 (144)		156 (37)		2124 (125)		77 (33)	
HMC	11325 (915)		830 (269)		12556 (661)		293 (137)	
SMMALA	9592 (2052)		61 (23)		10241 (2672)		47 (17)	
	$1\|2,3$		$2\|1\|3$		$1\|2\|3$		$2\|1\|3$	

by fixing $\boldsymbol{\theta}$ to the values used to generate the data. We notice that different likelihood functions heavily affect efficiency and convergence speed; in the examples considered here, the results show that in `LRG` it is possible to achieve efficiency one order of magnitude higher than in other models. The scaled versions of MH work well in the case of `LRG` (MH v1 is slightly better than MH v2), but do not offer guarantees of convergence on other models. ELL-SS achieves better efficiency and convergence than the scaled versions of MH. SMMALA, which uses gradient and curvature information, achieves good efficiency and faster convergence than MH v1, MH v2, and ELL-SS, but at the cost of one operation in $O(n^3)$ at each iteration, as the metric tensor is a function of $\mathbf{f}$ and needs to be factorized at each iteration. Overall, the results suggest that the scaled versions of HMC are the best sampling methods for $\mathbf{f}|\boldsymbol{\theta}, \mathbf{y}$. HMC v1 is slightly better than HMC v2, but it requires one extra inversion and one extra Cholesky decomposition compared to HMC v2 that does not require any operations in $O(n^3)$ once the covariance matrix of the GP is factorized.

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

In this section we present results about the sampling of hyper-parameters from the posterior distribution $\boldsymbol{\theta}|\mathbf{f}, \mathbf{y}$ which, given the hierarchical structure of the model, is simply $\boldsymbol{\theta}|\mathbf{f}$ independent from the data model. As reported in Tab. 1, the complexity of applying SMMALA and HMC is quite high compared to MH. MH requires one Cholesky factorization of Q at each iteration. In HMC, at each leapfrog step, the gradients of Q with respect to $\boldsymbol{\theta}$ are needed and the cheapest way to do this is by inverting Q first and noticing that all the remaining operations are in $O(n^2)$; this is done $\bar{\lambda}$ times on average at every iteration of HMC. Similarly, in SMMALA the gradient can be computed by inverting Q first; by doing so, the metric tensor can then be computed by d multiplications with the derivatives of Q and no other $O(n^3)$ operations.

The results are reported in Tab. 3, and were obtained by fixing $\mathbf{f}$ to the value used to generate the data and sampling only the length-scale parameters, as σ can be efficiently sampled using exact Gibbs steps. HMC improves quite substantially on efficiency, but not on speed of convergence; it may be worth employing some rescaling of the hyper-parameters to improve on this as suggested by Neal (1996). The performance of SMMALA is highly variable in efficiency and it converges more slowly than MH and HMC. This might be due to the skewness of the target

Table 4 Comparison of transition operators to sample $\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\nu}$ for data generated from models with four different likelihoods. Minimum ESS is computed as the average over 10 chains for 10 different data sets for each value of n and d. The last row in each sub-table reports the result of the statistical test to assess which operators achieve significantly faster convergence.

`LRG`

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH	556 (201)	▥	131 (33)	▥	512 (177)	▥	56 (11)	▥
HMC	2572 (1382)	▥	859 (278)	▥	2666 (973)	▥	223 (39)	◧
SMMALA	3833 (2032)	▥	65 (42)	▥	6877 (1584)	▥	47 (21)	◧
	$1, 3\|2$		$2\|1\|3$		$1\|3\|2$		$1\|2, 3$	

`LCX`

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH	818 (386)	▥	6 (4)	◧	1030 (397)	◧	3 (1)	◧
HMC	5169 (3297)	◧	11 (8)	◧	7145 (3852)	◧	4 (3)	◧
SMMALA	6158 (2788)	▥	9 (6)	■	8377 (1815)	◧	6 (4)	■
	$3\|1, 2$		$1, 2\|3$		$1, 2, 3$		$1, 2\|3$	

`VLT`

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH	859 (318)	◧	22 (6)	◧	795 (270)	■	8 (6)	◧
HMC	5680 (2634)	◧	48 (20)	◧	5233 (2482)	■	11 (11)	■
SMMALA	6274 (1896)	◧	14 (9)	◧	6950 (2763)	■	11 (9)	■
	$1, 2, 3$		$1\|2\|3$		$1, 2, 3$		$1\|2\|3$	

`ORD`

	$n = 100$				$n = 400$			
	$d = 2$		$d = 10$		$d = 2$		$d = 10$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
MH	689 (159)	▥	14 (7)	◧	552 (168)	◧	9 (6)	■
HMC	155 (296)	■	14 (11)	■	79 (115)	■	4 (4)	■
SMMALA	3356 (1661)	▥	11 (8)	◧	2328 (1423)	◧	19 (27)	◧
	$1, 3\|2$		$1, 3\|2$		$1, 3\|2$		$1, 3\|2$	

distribution, that is known to affect the efficiency of SMMALA (Stathopoulos and Filippone, 2011). The results indicate that MH strikes a good balance between efficiency and computational cost.

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

In this section we present the sampling of the hyper-parameters from the posterior distribution $\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\nu}$, where we fixed $\boldsymbol{\nu}$ to the values used to generate the data. The analysis of complexity shows that MH requires one Cholesky factorization at each iteration. In HMC, each leapfrog requires computing L and the gradient of L with respect to $\boldsymbol{\theta}$ and no other operations in $O(n^3)$; this can be computed using the differentiation of the Cholesky algorithm which requires d operations in $O(n^3)$ (Smith, 1995). Likewise, for SMMALA L and the d derivatives of L with respect to $\boldsymbol{\theta}$ are the only operations in $O(n^3)$ needed.

Table 5 Comparison of different strategies to sample $\mathbf{f}, \boldsymbol{\theta}|\mathbf{y}$ for data generated from a `LRG` model. The rightmost column reports the complexity of the different methods with respect to number of inversion and Cholesky decompositions. In KHR, $\bar{\kappa}$ represents the average number of iterations to run the Laplace Approximation.

	$n = 100$				$n = 400$					
	$d = 2$		$d = 10$		$d = 2$		$d = 10$			
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	#I	#C
AA	131(57)	▯▯▯▯	117(34)	▯▯▯▯	94(38)	▯▯▯▯	47(17)	▃▯▯▯	0	T
ASIS	138(63)	▯▯▯▯	168(49)	▯▯▯▯	98(39)	▯▯▯▯	60(25)	▯▯▯▯	0	$2T$
KHR	856(360)	▯▯▯▯	177(48)	▯▯▯▯	481(219)	▯▯▯▯	116(32)	▯▯▯▯	0	$\bar{\kappa}T + 2T$
SA	8(6)	▆▆▆▆	59(18)	▯▯▯▯	5(2)	█▆▆▆	14(6)	▂▂▂▯	0	T
SURR	173(95)	▯▯▯▯	90(32)	▯▯▯▯	157(51)	▯▯▯▯	35(15)	▂▂▂▯	T	$2T$
	$3\|5\|1, 2\|4$		$1, 2\|3, 4, 5$		$3\|5\|1, 2\|4$		$2, 3\|1\|4, 5$			

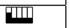
The results can be found in Tab. 4 and are again variable across different models. In general SMMALA and HMC do not seem to offer faster convergence with respect to the MH transition operator which is therefore competitive in terms of efficiency relative to computational cost.

4.3 Assessing the efficiency of different parameterizations

After analyzing the results in the previous section, we decided to combine the transition operators which achieved a good sampling efficiency with relatively low computational cost and ease of implementation. We decided that a good combination to be used in AA, SA, ASIS, and SURR could be as follows: sampling $\mathbf{f}$ using HMC v2 and $\boldsymbol{\theta}$ using MH; HMC v2 and MH where adapted during the burn-in phase and in HMC v2 we set the maximum number of leapfrog steps to 10. For the sake of brevity, we focus on the `LRG` model only; the results on efficiency and speed of convergence in sampling hyper-parameters are reported in Tab. 5.

It is striking to see how challenging it is to efficiently sample from the posterior distribution of latent variables and hyper-parameters. Sampling efficiency is generally low; this is consistent with our experience in other applications involving sampling in hierarchical models (Filippone et al., 2012). As expected, the SA parameterization is the worst among the ones we tested. The AA parameterization, ASIS, and SURR generally offer good guarantees of convergence within a few thousand iterations. SURR seems to be superior in efficiency, which is consistent with what reported in Murray and Adams (2010), but it requires more operations in $O(n^3)$ compared to AA and ASIS. ASIS slightly improves efficiency and speed of convergence with respect to the AA scheme but requires double the number of operations in $O(n^3)$. KHR seems effective in breaking the correlation between the two groups of variables, but it may require several iterations within the approximation used to sample $\mathbf{f}$. In the experiments considered here $\bar{\kappa}$ is around 8, so the best compromise between computations and efficiency seems to be given by the AA and ASIS parameterizations.

Table 6 Comparison of different strategies to sample $\mathbf{f}, \boldsymbol{\theta} | \mathbf{y}$ in four UCI data sets modeled using a `LRG` model.

	Pima $n = 768, d = 8$		Wisconsin $n = 683, d = 9$		SPECT $n = 80, d = 22$		Ionosphere $n = 351, d = 34$	
	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$	ESS	$\hat{R}$
AA	34 (4)		42 (15)		99 (18)		12 (5)	
ASIS	35 (8)		47 (11)		215 (23)		24 (8)	
KHR	153 (14)		20 (10)		101 (16)		2 (2)	
SA	5 (2)		7 (3)		97 (12)		11 (7)	
SURR	76 (10)		25 (14)		84 (14)		9 (4)	

5 Results on real data

We repeated the comparison of different parameterizations on four UCI data sets (Asuncion and Newman, 2007), namely the Pima, Wisconsin, SPECT, and Ionosphere data sets, which we modeled using `LRG` models; the results are reported in Tab. 6. We used the same priors and experimental setup as in the previous sections, except that all features were transformed to have zero mean and unit standard deviation, and latent variables were sampled iterating five updates of HMC v2. Also, chains were initialized sampling from the prior. Again, the SA parameterization shows the poorest efficiency and convergence speed, and the AA parameterization improves on that. Combining the AA and SA parameterizations using ASIS slightly improves on the AA parameterization, although the improvement is not dramatic. The SURR method improves on the AA parameterization, which is consistent with what reported in Murray and Adams (2010). The results of KHR are highly variable across data sets; in cases where the approximation to sample latent variables is accurate, the chains mix well. In some cases, however, the approximation is not accurate enough to guarantee a good acceptance rate, and the chains can spend a long time in the same position before accepting the joint proposal.

6 Conclusions

In this paper we studied and compared a number of state-of-the-art strategies to carry out the fully Bayesian treatment of GP models. We focused on four GP models and performed an extensive evaluation of efficiency, convergence speed, and computational complexity of several transition operators and sampling strategies.

The results in this paper show that latent variables can be sampled quite efficiently with little computational effort once the GP covariance matrix is factorized. This can be achieved by a simple variant of HMC that we introduced in this paper. About sampling hyper-parameters in different parameterizations, the results presented here indicate that the gain in sampling efficiency given by the use of complicated proposal mechanisms does not scale as much as their computational cost. It would be interesting to investigate some recently proposed variants to slice sampling (Thompson and Neal, 2010) and Hybrid Monte Carlo (Hoffman and Gelman, 2012) on the sampling of hyper-parameters.

The analysis of the results obtained by different parameterization suggest that AA is a sensible and computationally cheap parameterization with good convergence properties. AA performs similarly to ASIS at half the computational cost. It makes sense, however, to employ ASIS when in doubt about the best parameterization to use, although GP models with full covariance matrices will generally fall into the weak data limit as the $O(n^2)$ space and $O(n^3)$ time complexities constrain the number of data that can be processed.

In general, the results show how challenging it is to efficiently sample from the posterior distribution of latent variables and hyper-parameters in GP models and motivates further research into methods to do this efficiently. Some sampling strategies, such as the one based on the AA parameterization, are capable of achieving convergence within a reasonable number of iterations, and this makes it possible to carry out the fully Bayesian treatment of GP models dealing with a small to moderate number of samples. We have recently demonstrated that this is indeed the case in Filippone et al. (2012), but more needs to be done in the direction of developing robust stochastic based inference methods for GP models.

It would be interesting to investigate how performance is affected by the choice of the design, which in the simulated data presented here was assumed uniform. Also, we studied in particular GP models with the squared exponential ARD covariance function. It would be interesting to compare the method considered here in models characterized by other covariance functions, such as the Matérn, or sparse inverse covariance functions as in Rue et al. (2009); the latter would make it possible to test the strong data limit case. Finally, in this study we have not included a mean function for the GP prior or extra parameters for the likelihood function. This would require including the sampling of other quantities that may further impact on efficiency and speed of convergence.

References

1. Amari, S. and H. Nagaoka (2000). *Methods of Information Geometry*, Volume 191 of *Translations of Mathematical monographs*. Oxford University Press.
2. Asuncion, A. and D. J. Newman (2007). UCI machine learning repository.
3. Christensen, O. F., G. O. Roberts, and J. S. Rosenthal (2005). Scaling limits for the transient phase of local MetropolisHastings algorithms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 253–268.
4. Chu, W. and Z. Ghahramani (2005). Gaussian Processes for Ordinal Regression. *Journal of Machine Learning Research 6*, 1019–1041.
5. Cseke, B. and T. Heskes (2011). Approximate Marginals in Latent Gaussian Models. *Journal of Machine Learning Research 12*, 417–454.
6. Filippone, M., A. F. Marquand, C. R. V. Blain, S. C. R. Williams, J. Mourão-Miranda, and M. Girolami (2012). Probabilistic Prediction of Neurological Disorders with a Statistical Assessment of Neuroimaging Data Modalities. *Annals of Applied Statistics 6*(4), 1883–1905.
7. Filippone, M., M. Zhong, and M. Girolami (2012). On the fully Bayesian treatment of latent Gaussian models using stochastic simulations. Technical Report TR-2012-329, School of Computing Science, University of Glasgow.
8. Flegal, J. M., M. Haran, and G. L. Jones (2007). Markov Chain Monte Carlo: Can We Trust the Third Significant Figure? *Statistical Science 23*(2), 250–260.
9. Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical Science 7*(4), 457–472.

10. Girolami, M. and B. Calderhead (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(2), 123–214.
11. Hoffman, M. D. and A. Gelman (2012). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, to appear.
12. Knorr-Held, L. and H. Rue (2002). On Block Updating in Markov Random Field Models for Disease Mapping. *Scandinavian Journal of Statistics 29*(4), 597–614.
13. Kuss, M. and C. E. Rasmussen (2005). Assessing Approximate Inference for Binary Gaussian Process Classification. *Journal of Machine Learning Research 6*, 1679–1704.
14. Mackay, D. J. C. (1994). Bayesian methods for backpropagation networks. In E. Domany, J. L. van Hemmen, and K. Schulten (Eds.), *Models of Neural Networks III*, Chapter 6, pp. 211–254. Springer.
15. Minka, T. P. (2001). Expectation Propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, San Francisco, CA, USA, pp. 362–369. Morgan Kaufmann Publishers Inc.
16. Møller, J., A. R. Syversveen, and R. P. Waagepetersen (1998). Log Gaussian Cox Processes. *Scandinavian Journal of Statistics 25*(3), 451–482.
17. Murray, I. and R. P. Adams (2010). Slice sampling covariance hyperparameters of latent Gaussian models. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.), *NIPS*, pp. 1732–1740. Curran Associates, Inc.
18. Murray, I., R. P. Adams, and D. J. C. MacKay (2010). Elliptical slice sampling. *Journal of Machine Learning Research - Proceedings Track 9*, 541–548.
19. Neal, R. (2003). Slice Sampling. *Annals of Statistics 31*, 705–767.
20. Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.
21. Neal, R. M. (1996). *Bayesian Learning for Neural Networks (Lecture Notes in Statistics)* (1 ed.). Springer.
22. Neal, R. M. (1999). Regression and classification using Gaussian process priors (with discussion). *Bayesian Statistics 6*, 475–501.
23. Opper, M. and O. Winther (2000). Gaussian processes for classification: Mean-field algorithms. *Neural Computation 12*(11), 2655–2684.
24. Rasmussen, C. E. and C. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.
25. Robert, C. P. and G. Casella (2005). *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
26. Roberts, G. O. and O. Stramer (2002). Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodology and Computing in Applied Probability 4*(4), 337–357.
27. Rue, H., S. Martino, and N. Chopin (2009). 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.
28. Smith, S. P. (1995). Differentiation of the Cholesky Algorithm. *Journal of Computational and Graphical Statistics 4*(2), 134–147.

29. Stathopoulos, V. and M. Filippone (2011). 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), 167–168.
30. Thompson, M. and R. M. Neal (2010). Covariance-Adaptive Slice Sampling. Technical Report 1002, Department of Statistics, University of Toronto.
31. Tierney, L. and J. B. Kadane (1986). Accurate Approximations for Posterior Moments and Marginal Densities. *Journal of the American Statistical Association 81*(393), 82–86.
32. Vanhatalo, J. and A. Vehtari (2007). Sparse Log Gaussian Processes via MCMC for Spatial Epidemiology. *Journal of Machine Learning Research - Proceedings Track 1*, 73–89.
33. Wilson, A. G. and Z. Ghahramani (2010). Copula Processes. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.), *NIPS*, pp. 2460–2468. Curran Associates, Inc.
34. Yu, Y. and X.-L. Meng (2011). 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 20*(3), 531–570.

A SA and AA parameterizations

A.1 Sufficient Augmentation (SA)

We derive here the quantities needed to apply the transition operators considered in this work in the SA parameterization. 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 σ could be marginalized out, but it would not be possible to get manageable expressions for the metric tensor with respect to $\boldsymbol{\tau}$; for $\mathbf{f}$, instead, this would be possible. We do not pursue this here, and we leave it for future investigation.

By inspecting the log-joint density, we see that we can obtain the conditional density for σ in the following 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 on σ in the form $\mathrm{invGa}(\sigma|a, b)$ with shape a and scale b, we can sample directly:

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

The gradients of the log-joint density needed to apply gradient based operators are:

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

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

The FI for latent variables and parameters are:

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

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

Given that the likelihood factorizes with respect to the observations, the Hessian of $\mathcal{L}$ with respect to $\mathbf{f}$ is diagonal, so $R = \mathrm{FI}_{\mathbf{f},\mathbf{f}}$ is diagonal as well. The metric tensors are the FI matrices plus the negative Hessian of the priors:

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

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

A.2 Ancillary Augmentation (AA)

We derive here the quantities needed to apply the transition operators considered in this work in the AA parameterization. The expression of the log-joint density is the same as in the SA case, bearing in mind the transformation $\mathbf{f} = \sqrt{\sigma} L \boldsymbol{\nu}$; this yields:

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

The gradient with respect to the hyper-parameters can be computed by using the chain rule of derivation and standard properties of derivatives of vector valued functions:

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

The FI matrix is readily obtained as:

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

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

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

B GP models considered in this paper

B.1 Logistic regression with GP priors (LRG)

Let:

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

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

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

The gradient with respect to $\mathbf{f}$ results in:

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

The computation of diagonal elements of the FI matrix for $\mathbf{f}$ requires the expectations of y_i^2 which are the same as the expectations of y_i, that are l_i^+; this leads to $R_{ii} = l^+(f_i)l^-(f_i)$.

B.2 Log-Gaussian Cox model (LCX)

In this model, observations follow a Poisson distribution with mean computed as an exponentially transformed version of the corresponding latent variables:

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

The gradient with respect to $\mathbf{f}$ and the diagonal elements of R result in:

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

B.3 Stochastic Volatility model with GP priors (VLT)

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

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

The gradient with respect to $\mathbf{f}$ and the diagonal elements of R result in:

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

B.4 Ordinal Regression with GP priors (ORD)

In this model, latent variables are thresholded at r points that will be denoted by $b_0, \ldots, b_r$, with $b_0 = -\infty$ and $b_r = +\infty$. Then, y is the index of the interval where the corresponding latent variable f falls. The likelihood of an observed label y_i associated to the ith latent variable 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 variables 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:

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

In particular:

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

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

By writing the diagonal elements of Hessian of the log-likelihood computed for $y_i = s$

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

it is possible to compute the expectation of the negative Hessian as:

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

Note that the formulation in this paper is slightly different from the one in Chu and Ghahramani (2005), where σ is dropped and thresholds are inferred instead.