NeuroImage 118 (2015) 133–145

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg

Inversion of hierarchical Bayesian models using Gaussian processes

Ekaterina I. Lomakina [a,b,*], Saee Paliwal [b], Andreea O. Diaconescu [b], Kay H. Brodersen [a,b], Eduardo A. Aponte [b], Joachim M. Buhmann [a], Klaas E. Stephan [b,c]

[a] Department of Computer Science, ETH Zurich, Switzerland
[b] Translational Neuromodeling Unit (TNU), Institute for Biomedical Engineering, University of Zurich & ETH Zurich, Switzerland
[c] Wellcome Trust Centre for Neuroimaging, University College London, United Kingdom

ARTICLE INFO

Article history:
Received 1 October 2014
Accepted 29 May 2015
Available online 3 June 2015

Keywords:
Global optimisation
Gaussian processes
Bayesian inference
MCMC
Dynamic causal modelling
Hierarchical models

ABSTRACT

Over the past decade, computational approaches to neuroimaging have increasingly made use of hierarchical Bayesian models (HBMs), either for inferring on physiological mechanisms underlying fMRI data (e.g., dynamic causal modelling, DCM) or for deriving computational trajectories (from behavioural data) which serve as regressors in general linear models. However, an unresolved problem is that standard methods for inverting the hierarchical Bayesian model are either very slow, e.g. Markov Chain Monte Carlo Methods (MCMC), or are vulnerable to local minima in non-convex optimisation problems, such as variational Bayes (VB). This article considers Gaussian process optimisation (GPO) as an alternative approach for global optimisation of sufficiently smooth and efficiently evaluable objective functions. GPO avoids being trapped in local extrema and can be computationally much more efficient than MCMC. Here, we examine the benefits of GPO for inverting HBMs commonly used in neuroimaging, including DCM for fMRI and the Hierarchical Gaussian Filter (HGF). Importantly, to achieve computational efficiency despite high-dimensional optimisation problems, we introduce a novel combination of GPO and local gradient-based search methods. The utility of this GPO implementation for DCM and HGF is evaluated against MCMC and VB, using both synthetic data from simulations and empirical data. Our results demonstrate that GPO provides parameter estimates with equivalent or better accuracy than the other techniques, but at a fraction of the computational cost required for MCMC. We anticipate that GPO will prove useful for robust and efficient inversion of high-dimensional and nonlinear models of neuroimaging data.

Introduction

The application of neuroimaging methods, such as functional magnetic resonance imaging (fMRI), to cognitive neuroscience has significantly enhanced our understanding of brain function. Following the advent of fMRI in the early 1990s, a first decade of neuroimaging research focused on the problem of mapping, i.e., *where* particular cognitive functions are implemented, by localising task-induced activity in specific brain structures. Approximately one decade later, this perspective began to shift towards a computational approach, emphasising the question *how* cognitive functions are implemented. This was accomplished by introducing a variety of mathematical models to neuroimaging (for review, see Friston and Dolan, 2010). These models can be categorised into two groups. First, models of information processing ("computational models") from reinforcement learning or Bayesian accounts of cognition which serve to infer, from the observed subject-specific behaviour, trajectories of computational quantities such as prediction errors. These computational quantities can then be used as regressors in conventional mass-univariate analyses based on the general linear model (GLM). This approach, often referred to as "model-based fMRI" (Glascher and O'Doherty, 2010), has been very successful in providing a more fine-grained account of the computations implemented in particular brain regions. Prominent examples include the encoding of different types of prediction errors and uncertainty in brain regions such as the dopaminergic midbrain (D'Ardenne et al., 2008), striatum (O'Doherty et al., 2003), or the basal forebrain (Iglesias et al., 2013), and cortical structures like the insula (Preuschoff et al., 2008). Over the years, the models employed have become increasingly complex, from classical reinforcement learning models such as the Rescorla-Wagner model (Rescorla and Wagner, 1972) or temporal difference learning (Schultz et al., 1997) to more complicated hierarchical models (e.g., Behrens et al., 2007). One particular model we will examine in more detail below, the Hierarchical Gaussian Filter (HGF; Mathys et al., 2011), describes a hierarchy of coupled belief updating processes whose subject-specific parameters are inferred through variational inversion.

The second group of models concerns the physiological processes underlying neuroimaging data. In particular, these are dynamic causal models (DCMs) of fMRI (Friston et al., 2003) or electrophysiological data (David et al., 2006). As generative models, DCMs embody a

[*] Corresponding author at: Wilfriedstrasse 6, 8032 Zurich, Switzerland. Fax: +41 44 634 91 31.
E-mail address: ekaterina.lomakina@inf.ethz.ch (E.I. Lomakina).

http://dx.doi.org/10.1016/j.neuroimage.2015.05.084
1053-8119/© 2015 Elsevier Inc. All rights reserved.

probabilistic forward mapping from hidden brain states (neuronal and haemodynamic states) to the observed measurements in multiple interacting regions. A Bayesian inversion of this forward model then enables computing of the posterior probability of the parameters which represent, for example, effective connectivity strengths. An additional key goal is to obtain an approximation to the log-evidence in order to perform model selection, i.e., comparing alternative explanations of how the observed data could have been caused (Penny et al., 2004).

Many of the computational and physiological models mentioned above, e.g. DCM and HGF, can be understood as special cases of hierarchical Bayesian models (HBMs). These models are powerful tools for analysing behavioural and neuroimaging data and are finding widespread application. One potential problem is, however, that statistical inference can be difficult due to the computational challenges of model inversion (for an earlier discussion focused on DCM, see Daunizeau et al., 2011). Commonly employed inversion methods are either very slow, e.g. Markov Chain Monte Carlo Methods (MCMC) such as the Metropolis–Hastings algorithm (MH; Metropolis and Ulam, 1949), or are susceptible to local minima, such as gradient descent schemes used in variational Bayesian methods (Friston et al., 2007; Daunizeau et al., 2014).

Here, we consider Gaussian process optimisation (GPO; Osborne et al., 2009; Frean and Boyle, 2008) as an alternative to MCMC and variational methods. GPO offers three potential advantages: (i) as a global optimisation method for sufficiently smooth and efficiently evaluable objective functions it is less susceptible to local minima than gradient descent schemes; (ii) it makes nonparametric assumptions about the objective functions; and (iii) it is computationally more efficient than MCMC.

Specifically, in this technical note, we propose an implementation of GPO for inverting HBMs commonly used in neuroimaging, including DCM for fMRI and a standard three-level HGF for learning and decision-making tasks. Critically, to deal with the “curse of dimensionality” (Bellman, 1957) in applications where the number of parameters to optimise over is fairly high (e.g., >30 parameters as often encountered in DCMs), we introduce a novel combination of GPO and local gradient methods which greatly reduces procedural complexity and ensures computational tractability. The utility of this implementation for DCM and HGF is evaluated using both synthetic data (from simulations) and empirical data.

The paper is structured as follows. The “Methods” section describes the idea of our method along with a brief overview of Gaussian Processes and a number of related technical issues, such as optimisation of hyperparameters and the challenge by the ‘curse of dimensionality’. While GPs have found application for classification analyses of neuroimaging data (e.g., Marquand et al., 2010; Salimi-Khorshidi et al., 2011; Mourão-Miranda et al., 2012; Pyka et al., 2013), they have found little application in computational modelling of brain physiology or cognition so far. We therefore discuss some of their central properties in a tutorial-like fashion. In the “Results” section we present the performance of our method (i) in finding the global maximum of a highly multimodal function, (ii) for inverting DCMs given synthetic data (and thus known underlying parameters), and (iii) for inverting HGFs given real data. When inverting these models, we compare the performance of GPO to two well-established approaches: Gauss–Newton descent (in the context of the Variational Laplace scheme typically used for DCMs; Friston et al., 2007), and an MH sampling scheme.

Methods

Global optimisation of complicated objective functions

As a basis for our implementation described below, this section briefly summarises a general approach to optimising multimodal functions in cases where stochastic optimisation methods such as the MH algorithm are computationally too expensive while local methods such as Gauss–Newton descent schemes are too vulnerable to local minima. This general framework, Bayesian Global Optimisation (BGO; Mockus et al., 1978), offers a useful compromise between MH and local methods. The underlying idea of BGO is to approximate the target function with some easy-to-evaluate proxy based on a set of points over which the target function has already been evaluated, and optimise the current approximation instead of the target function itself. The key challenge here is to compute, based on a few evaluated data points, a (possibly multimodal) approximation function which allows for rapid identification of a candidate optimum, while suggesting additional data points whose evaluation may help to further improve the present estimate. More specifically, the approximation is derived from a Gaussian process which, given a set of evaluated points and under the assumption of the function being smooth, predicts how the function typically behaves over its domain. Critically, since this approach returns confidence intervals for function values at non-sampled points, it enables us to derive a number of criteria which guide the sampling from hitherto unexplored domains, such as Expected Improvement (Mockus et al., 1978) or Upper Confidence Bound (Srinivas et al., 2010). These criteria suggest a principled exploration-exploitation trade-off. An outline of the general BGO algorithm is provided in Box 1.

Notably, the approximating function can be any regression function, including ridge regression, random forest or Gaussian processes (GP). In this paper we will focus on global optimisation using GP as it has several important advantages: (i) GPs are fast at evaluation; (ii) their only assumption is that the underlying function is structurally smooth in some space; (iii) by intuitive adjustments, GP can approximate a wide range of functions; finally and most importantly, (iv) GP has an inbuilt approximation not only to the function itself but also to the variance of this approximation. Before we explain these points and their importance for the overall scheme in more detail below, it should be emphasised that using GP for BGO is well-established. For example, using GP mean estimates for approximation corresponds to “kriging” (Krige, 1951). By contrast, more recent approaches to efficient global optimisation exploit both mean and variance estimates provided by GP (Osborne et al., 2009).

Gaussian processes

This section provides a short, tutorial on Gaussian processes. A Gaussian process is defined as a stochastic process for which realisations are jointly distributed according to the multivariate normal distribution. More specifically, a collection of N points and their corresponding targets are jointly distributed according to the following normal multivariate distribution:

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \cdots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \ldots & k(x_N,x_N) \end{bmatrix}\right), \tag{1}$$

Box 1
Outline of a general algorithm for Bayesian Global Optimisation.

- Define initial points $\boldsymbol{x}_{init}$
- Evaluate the target function f in $\boldsymbol{x}_{init}$ and store $\boldsymbol{x}_{init}$ and $f(\boldsymbol{x}_{init})$ in the approximation set $S = [x_{init}; f(x_{init})]$
- Until expected improvement E[I] falls below a pre-defined threshold:
  - ○ Approximate S with the approximation function f^a
  - ○ Find new point which maximises expected improvement: $x_{new} = \text{argmax}_{x^*} E[I(x^*)]$
  - ○ $S = [S \{x_{new}; f(x_{new})\}]$.

where $\boldsymbol{x} = \{x_1, x_2, \cdots, x_N\}$ is a collection of points, $\boldsymbol{t} = \{t, t_2, \cdots, t_N\}$ is a collection of corresponding targets, and $k(\cdot,\cdot)$ is some kernel or covariance function. The meaning of the kernel function will be explained in more detail below. For the moment it is sufficient to note that the kernel function encodes the smoothness of predictions, i.e., non-zero off-diagonal elements of the covariance function enforce that data points close to each other result in similar target variables. In other words, we can think of distribution (1) as a prior distribution over the function t(x), which forces it to be smooth.

Analogously we can specify a joint distribution over a collection of already observed data points and previously unseen ones:

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \\ t^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots & k(x_1,x_N) & k(x_1,x^*) \\ k(x_2,x_1) & k(x_2,x_2) & \cdots & k(x_2,x_N) & k(x_2,x^*) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \cdots & k(x_N,x_N) & k(x_N,x^*) \\ k(x^*,x_1) & k(x^*,x_1) & \cdots & k(x^*,x_N) & k(x^*,x^*) \end{bmatrix}\right). \tag{2}$$

Here and in the following derivations we deal with the case of a single unseen data point $\{x^*, t^*\}$ but this easily generalises to a collection of unseen points.

Given distribution (2), one can derive the conditional distribution $p(t^*|\boldsymbol{t})$. Using a Schur complement, one can show that this distribution will be a multivariate normal distribution as well, with the following mean and variance:

$$\begin{aligned} \mu_y(x^*) &= \mu_y(x^*) = \boldsymbol{k}^T K_N^{-1} \boldsymbol{t} \\ \sigma_y^2(x^*) &= k(x^*,x^*) - \boldsymbol{k}^T K_N^{-1} \boldsymbol{k}, \end{aligned} \tag{3}$$

where K_N is the covariance matrix in Eq. (1) and $\boldsymbol{k} = [k(x^*, x_1), \cdots, k(x^*, x_N)]$. This means that the distribution over any new point can be fully estimated analytically and requires only one inversion of an $N \times N$ matrix. This computational simplicity renders GP an efficient prediction method, especially when a dataset is not too large. In addition, various numerical techniques exist which accelerate the inversion of the covariance matrix; others serve to increase its numerical stability, e.g., Cholesky decomposition (Rasmussen and Williams, 2006).

However, so far we have assumed that there is no noise in the model, in other words, that all targets are observed exactly. This assumption does not necessarily hold, as our observations can be corrupted due to measurement noise. To generalise the model, we can assume that the actually observed variables $\boldsymbol{y}$ are equal to true targets $\boldsymbol{t}$ corrupted by some normally distributed noise:

$$p(\boldsymbol{y}|\boldsymbol{t}) = N\left(t, \beta^{-1} I^{N \times N}\right), \tag{4}$$

where $I^{N \times N}$ is an $N \times N$ identity matrix, and β specifies the precision of the observations. From this the distribution over the target for the new point can be derived given noisy observations $\boldsymbol{y}$. This distribution is also Gaussian with the following mean and variance:

$$\begin{aligned} \mu_y(x^*) &= \boldsymbol{k}^T C_N^{-1} \boldsymbol{y} \\ \sigma_y^2(x^*) &= c - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k}, \end{aligned} \tag{5}$$

where

$$\begin{aligned} C_N &= K + \beta^{-1} I^{N \times N} \\ \boldsymbol{k} &= [k(x^*, x_1), \cdots, k(x^*, x_N)] \\ c &= k(x^*, x^*) + \beta^{-1}. \end{aligned} \tag{6}$$

For details of the derivation, please see Rasmussen and Williams (2006) who fully specify GP based prediction for the case of noisy observations. Importantly, the introduction of noise invokes a regularisation of the covariance matrix and thus makes numerical inversion of the matrix more stable.

Gaussian processes for global optimisation

Eq. (5) highlights that the computational efficiency of GP based prediction is limited by the efficiency of computing mean and variance as specified. From Eq. (5) we can see that this requires only one inversion of the covariance matrix C_N, where N is the number of observed points (typically much smaller than the number of points we want to test). All other computations include the evaluation of the kernel function $k(\cdot,\cdot)$ and matrix multiplication and thus grow linearly with the number of test points. In other words, at relatively low computational cost, one can evaluate as many test points as required, given a set of observed data.

The smoothness assumption made by GP follows directly from the kernel function in Eq. (1). This kernel function specifies the similarity between data points, as reflected by the off-diagonal elements of the covariance matrix. It means that targets of points close to each other under a chosen kernel function should also be similar, which conforms to an intuitive notion of smoothness. An illustration of this concept is depicted in Fig. 1. Importantly, a kernel function specifies a mapping

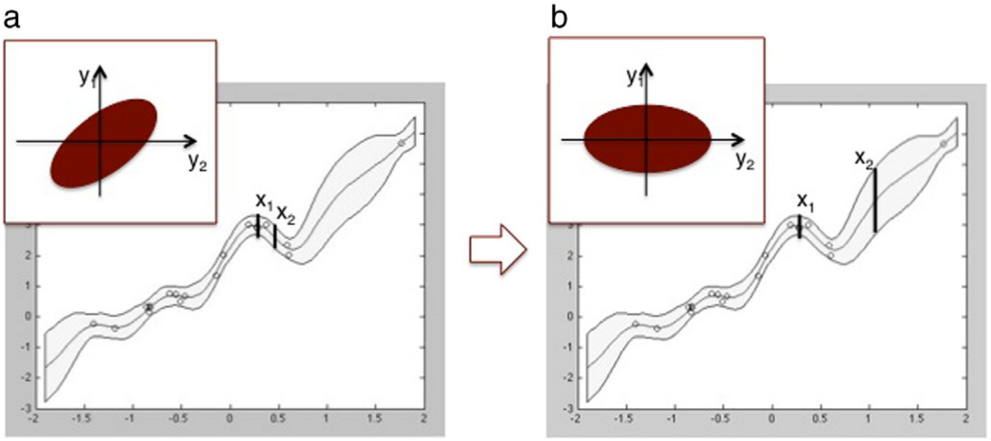

Fig. 1. An illustration of Gaussian processes and the underlying smoothness assumption about functions. Panel a demonstrates the situation when two points are close to each other. In this case, the covariance matrix of the joint distribution of their targets has large off-diagonal elements which enforces similarity in the value of the associated targets. Panel b presents the case where points are sufficiently far from each other that the associated targets exhibit only small dependencies.

of data points to a potentially unknown space in which this kernel function is an inner product, thus smoothness specified in this space does not necessarily imply smoothness in a Euclidian space. However, for typical kernel functions such as a squared exponential, this complication does not arise.

The central role of the kernel function for GP is a major reason for its flexibility. By varying the kernel function $k(\cdot,\cdot)$ and its hyperparameters, we obtain very different predictions, as illustrated in Fig. 2, and can thus approximate functions with various dynamical ranges or smoothness properties. Moreover, the method is not limited to real vectors but can work with any type of object, including graphs or distributions, as long as we can specify a meaningful kernel or similarity metric for them. However, this flexibility comes at a price, as the specification of the kernel function and its hyperparameters becomes critically important. We will discuss possible kernel functions and their properties in more detail below.

The fourth property of GP we highlighted, i.e., providing both an approximation and the variance of this approximation, follows directly from Eq. (5). This is crucially important for using GPs as a method to implement BGO as the variance estimate in Eq. (5) is the basis for assessing the "expected improvement" obtained by selecting new evaluation points. Importantly, in Eq. (5) the variance depends only on the pairwise similarities of the points and not on their values. Simply speaking, high variance of a candidate point for evaluation indicates that this point is distant from previously explored domains and may be close to a "hiding" extremum. Having access to both the approximation and its variance allows one to balance exploration against exploitation.

Given these additional properties, an optimisation algorithm based on GP has the following general structure described in Box 2:

There are number of possible criteria for convergence, i.e. Expected Improvement (cf. Mockus et al., 1978; Jones et al., 1998). However, in this work, we primarily focus on the *Upper Confidence Bound* (UCB) criterion (Srinivas et al., 2012) due to both its simplicity and robustness in practical applications:

$$UCB\left(\mu(x^*|S),\ \sigma^2(x^*|S)\right) = \mu(x^*|S) + \alpha\sigma^2(x^*|S), \tag{7}$$

where $\alpha \geq 0$ enables control of the exploration–exploitation trade-off. If α turns to zero, this approach reduces to using the approximation alone, whereas if α is greater than zero, the variance of the approximation is also taken into account, forcing the algorithm to visit unexplored areas of the domain of the approximating function, despite their predicted low function value. An illustration of this concept is presented in Fig. 3.

In this subsection we have illustrated some key properties of GP and explained how they can be useful for BGO. We now turn to a number of technical considerations caused by the nature of GP per se, or driven by the specific applications we have in mind, i.e., inversion of hierarchical Bayesian models for neuroimaging and behavioural data.

Kernel functions

The kernel function determines the space in which smoothness is assumed and thus becomes a crucial choice for approximation. Most kernel functions have their own hyperparameters, which can either be defined a priori by the user or optimised within the algorithm. These hyperparameters can strongly affect the behaviour of the kernel function, as demonstrated in Fig. 2.

Here, a kernel function is defined as a function of two arguments, $K(\cdot,\cdot)$, which represents an inner product of single argument functions $\varphi(\cdot)$ in some potentially infinite dimensional space V:

$$K(x_1, x_2) = \varphi(x_1), \varphi(x_2)_V. \tag{8}$$

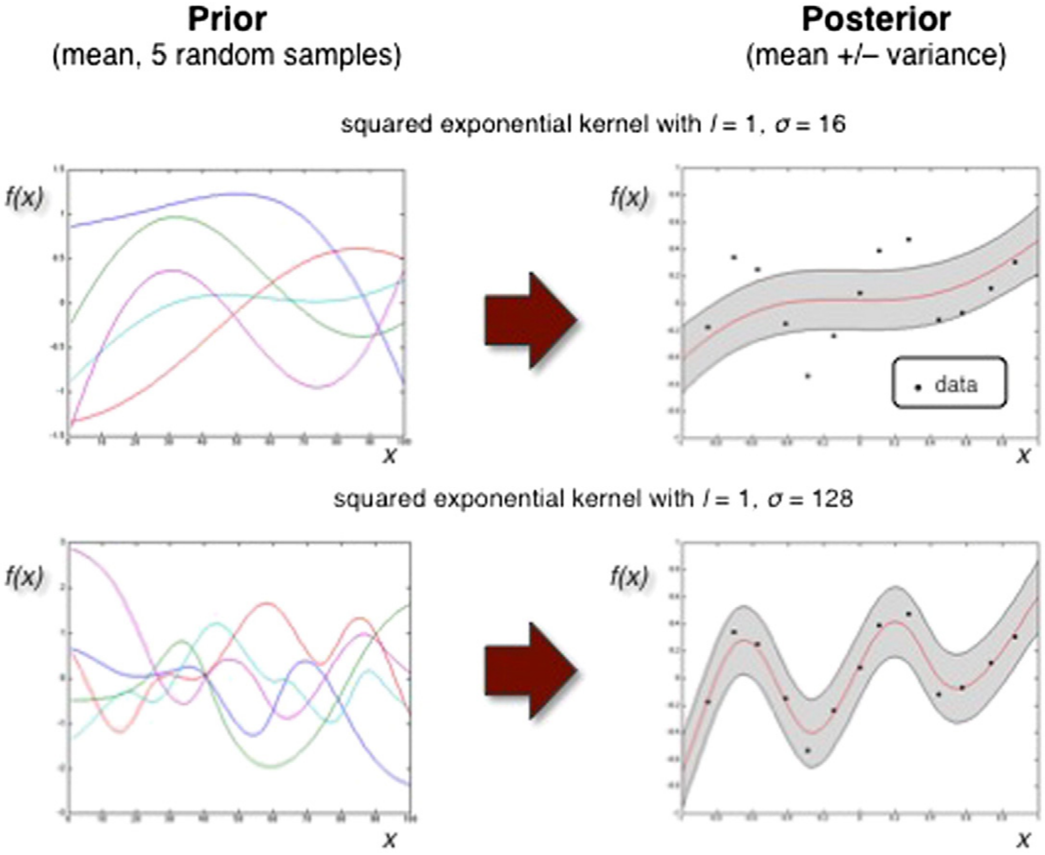

Fig. 2. An illustration of how different parameters of the kernel function translate into different priors over functions and thus lead to different posterior beliefs given the same data. In this particular example, a hyperparameter of the squared exponential kernel is changed (see Eq. (11)), resulting in different prior beliefs about the function's smoothness. Here, a less smooth function better matched the observed data.

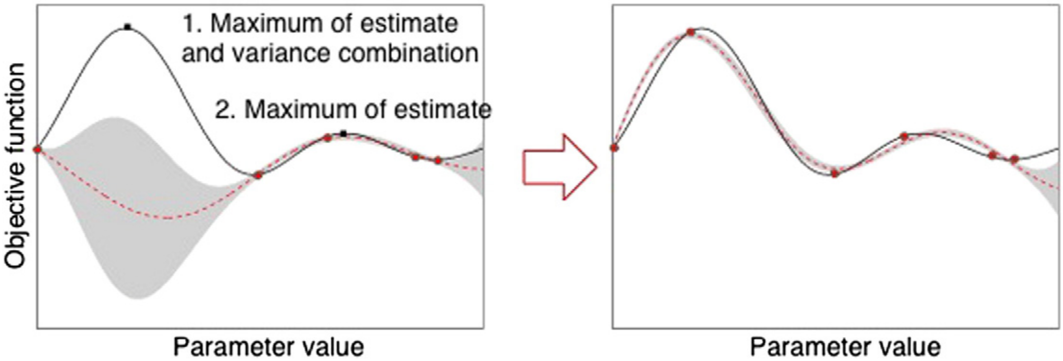

Fig. 3. An illustration of how Gaussian process optimisation utilises variance estimates of the approximating function (red line) to evaluate unexplored regions which promise to provide a maximum amount of additional information.

To prove that some function can be used as a kernel function we need to show that it can be represented in the form above. As *V* can be infinite dimensional, it is not always possible to express it explicitly; however, we can construct kernels from simpler ones using a set of rules. For example, a linear combinations of kernels is also a kernel (for further details, see Bishop, 2006). In the following we briefly discuss some of the most widely used kernels.

A *linear kernel* is the simplest possible kernel, in which we assume *V* to be the same as the kernel space. In other words, this kernel function is just a linear product of Euclidian vectors of data point features:

$$K(x_1, x_2) = x_1, x_2. \tag{9}$$

This kernel, however, is not appropriate for functions which are assumed to be multimodal and thus pose a highly nonlinear optimisation problem.

Another widely used kernel is the *Matern kernel*:

$$K(x_1, x_2) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x_1 - x_2|}{l} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{|x_1 - x_2|}{l} \right), \tag{10}$$

where K_V is a modified Bessel function of the second type and l, σ^2 and v are non-negative hyperparameters. The special case of $\nu \to \infty$ yields the widely used *Gaussian* (*squared exponential*) *kernel*:

$$K(x_1, x_2) = \sigma^2 \exp\left(\frac{-(x_1 - x_2)^2}{2l^2} \right). \tag{11}$$

Here, an intuitive interpretation of the hyperparameters is that *l* encodes the smoothness of the function (the larger *l*, the more distant data points will affect each other), and σ^2 specifies the overall amplitude of the function (see Fig. 2). It is important to note that *l*, in the case of a multivariate observation, can become a vector of the parameters and thus specify the relative sensitivity along each parameter.

The choice of a kernel function is a central issue for the application of Gaussian process. For an in-depth discussion, the reader is referred to Duvenaud (2014). In brief, sometimes the choice of kernel function can be guided by a priori knowledge; for example, if we know that the target function is periodic or linear, a periodic/linear kernel is a natural choice. Another approach is to select the kernel function from a set of standard options (as those described above) using cross-validation or Bayesian model selection. Alternatively, one can choose a kernel function which is sufficiently flexible that it can capture target functions with various smoothness under different regimes of hyperparameters, e.g. the Matern kernel, and infer appropriate hyperparameters from the data.

Inferring the hyperparameters, including the precision parameter β (see Eq. (4)) can exploit a priori knowledge about the target function, if available, or can be inferred using maximum likelihood estimation, given the GP approximation under a specific kernel function. Alternatively, optimisation of the hyperparameters could be performed online, for example, locally using a gradient descent method such as Gauss–Newton (cf. Rasmussen and Williams, 2006). Here, the trade-off between computational efficiency and approximation quality needs to be optimised. In the current work, we use the Matern kernel ($\nu = \frac{3}{2}$) for both real and synthetic data and perform a few iterations (up to 15 if convergence is not achieved earlier) of the local optimisation of the parameters at every approximation step, starting from their previous values.

Dynamical causal modelling for fMRI

DCM for fMRI is a generative model of the blood oxygen level dependent (BOLD) signal in multiple regions (Friston et al., 2003). It has a hierarchical structure with two layers. The evolutions of hidden neuronal states (one or several per region) which interact through synaptic connections are described by differential equations. These neuronal dynamics are translated into regionally specific BOLD signals through a hemodynamic forward model (Friston et al., 2000; Stephan et al., 2007). The DCM parameter set thus consists of three subsets: parameters of the neuronal state equations ("connectivity parameters"), parameters of the hemodynamic forward model ("hemodynamic parameters") and hyperparameters which encode the amplitude of observation noise in each region.

DCM is formulated in a fully Bayesian setting with Gaussian or log-normal priors on (hyper)parameters. Given a particular model, Bayesian inversion serves to obtain the posterior distributions of the parameters; typically these are specified in terms of maximum a posteriori (MAP) estimates and posterior variance. Additionally, as a basis for comparing alternative models, one would like to obtain an approximation to the log evidence (or log marginal likelihood) as a measure for the balance between the fit (accuracy) of the model and its complexity (Bayesian model selection, BMS; MacKay, 2003). However, as the model evidence is an integral over the joint probability, it is usually prohibitively expensive to compute. One solution to this problem is variational model inversion: by optimising a free energy bound on the log evidence, one implicitly diminishes the Kullback–Leibler divergence between an approximate posterior and the (unknown) true posterior. In other words, optimising the free energy bound both yields an approximation to the log evidence and the MAP estimates. In DCM, this variational inversion scheme is combined with a Laplace approximation (i.e., approximating the posterior by a Gaussian centred on its mode). Collectively, this has led to an efficient Bayesian inversion scheme called "Variational Laplace" (cf. Friston et al., 2007) and is used in the current implementation of DCM in the Statistical Parametric Mapping (SPM) software package (http://www.fil.ion.ucl.ac.uk/spm). While this method is computationally efficient and provides both MAP estimates of the parameters and the log evidence, it is susceptible to local minima, as it implicitly assumes a unimodal objective function (see Daunizeau et al., 2011 for a previous discussion of this potential problem).

In our approach, we use GP global optimisation to find the mode of the posterior distribution by maximising the log joint. To initialize the algorithm, we randomly sample K (typically 20 points) from a multivariate normal distribution (i.e., the priors of the model parameters) to induce a certain degree of exploration from the first step. Another initialization option is to sample from nodes of a predefined grid. This, however, is an approach that is applicable only to low-dimensional problems and becomes intractable in case of DCMs with a large number of free parameters. As DCMs often contain 30 or more parameters, parameter estimation is typically not a low-dimensional problem and raises certain challenges which we address below in the "Curse of dimensionality" section.

To evaluate our GP implementation for DCM, we compared GPO against the standard inversion scheme (Variational Laplace) for DCM in SPM, as well as against MCMC (i.e., the MH algorithm). To this end, we used a "ground truth" scenario, where synthetic data were generated from DCMs with known parameters, as described below.

Hierarchical Gaussian filtering (HGF)

As a complementary approach to assessing the accuracy of different optimisation methods for synthetic data, we now turn to the analysis of empirical data, using a different model than DCM. This is the Hierarchical Gaussian Filter (HGF), a hierarchical Bayesian model of learning which consists of a hierarchy of coupled Gaussian random walks. In the standard three-level HGF implementation (see Mathys et al., 2011), this coupling is specified by three parameters which encode a subject's individual approximation to Bayes optimal learning. A detailed account of this model can be found in Mathys et al. (2011); for a recent application to fMRI data, see Iglesias et al. (2013). Importantly, the standard implementation of the HGF rests on a quasi-Newton local gradient descent method (QN) for model inversion. This algorithm is fast and robust, but assumes that the objective function is convex. As a local method, it is vulnerable to local extrema.

In recent work, we adopted a similar evaluation approach as described for DCM above, i.e., we generated synthetic data and tested the accuracy of HGF parameter estimation for different optimisation methods, i.e., QN, GPO, and MH (Mathys et al., 2014). While this previous evaluation found that all three methods were comparably accurate, it concerned the standard implementation of a three-level HGF, where the assumption of a unimodal convex objective function is likely to hold. Present applications of the hierarchical Bayesian model of cognition move to increasingly more complex models with more complicated objective functions (e.g., Diaconescu et al., 2014). This increase in complexity raises the question whether the equivalence of optimisation methods still holds.

Here, we examine this issue, using a set of nine different HGFs, some of which are considerably more complex than previous implementations. The models are applied to empirically measured behavioural data from a social learning paradigm (Diaconescu et al., 2014). This paradigm examines how humans represent other agents' intentions in an interactive economic game, which included periods of both aligned and conflicting interests between subjects who were randomly assigned to a "player" or an "adviser" role. The key idea underlying the various models we tested was that participants employ hierarchically-structured learning as they infer on both the advice accuracy and the volatility of the adviser's changing intentions. This paradigm yields data which enable a challenging test for model inversion: (i) binary inputs and outputs but continuous hidden states, (ii) imbalanced input classes, (iii) heterogeneous player–adviser pairs, and (iv) complex learning processes under pronounced volatility (of the adviser's intentions).

As above, we chose the log joint probability as an objective function. We compared the performance of QN, GPO and MH with respect to the objective function values as well as to additional criteria such as model selection performance and predicting independent questionnaire scores of the subjects. The latter analysis uses an external criterion to assess the predictive validity of the model and how it is affected by the inversion scheme.

Curse of dimensionality

Typically, global optimisation by GPO is applied to active learning problems where the number of parameters is relatively small or the overall number of potential inputs is limited (for example see Krause et al., 2008; Vezhnevets et al., 2012; Osborne et al., 2009). However, in some of the problems mentioned above, the dimensionality of the parameter space can be quite high (e.g., DCMs for fMRI often have several dozen parameters). This poses some challenges which motivated the particular GPO implementation we propose in this paper.

The first challenge concerns the GP surface itself. As dimensionality increases, the number of evaluation points grows exponentially. Despite the computational efficiency of the evaluation of the upper confidence bound (Eq. (7)), this causes substantial computational problems very quickly, even when using a very rough grid. To avoid this situation, in our implementation the upper confidence bound was only computed for a fixed number of points (typically 10^6) which were randomly sampled from a large open sphere around the point of the current maximum. The radius of the sphere was taken to be two orders of magnitude larger than the variance of the parameters' priors (σ_θ) to guarantee coverage of the whole parameter space of interest.

The second challenge caused by high dimensionality concerns the target function optimisation. As the space grows exponentially with dimensionality, GP optimisation becomes much less computationally efficient than any local method (though still more efficient than sampling methods). To compensate for this, we departed from conventional GPO approaches and married local and global approaches, using the points identified by GP on the basis of the UCB criterion (Eq. (7)) for a subsequent local search (see Box 3 for details).

Practically, this meant that points chosen by GP optimisation were not only saved to the approximation set but also used as initialization points for a local gradient-based optimisation method, i.e., the quasi-Newton Broyden–Fletcher–Goldfarb–Shanno method (BFGS; Nocedal and Wright, 2006) which is computationally highly efficient. Since the initial point proposed by GP was likely to be in the vicinity of the actual extremum, we could afford restricting the local search to a few iterations and thus ensure high computational efficiency. In summary, combining global and local approaches in this way yields an excellent balance between precision and efficiency.

Comparision of GPO to others techniques

To evaluate its precision and efficiency, we compared our GPO implementation to other well-established inference schemes. Standard

Box 2
Outline of a general algorithm for BGO based on GP.

```
• Define initial points $\boldsymbol{x}_{init}$
• Evaluate the target function $f$ in $\boldsymbol{x}_{init}$ and store $\boldsymbol{x}_{init}$ and $f(\boldsymbol{x}_{init})$
  in the approximation set $S = [x_{init}; f(x_{init})]$
• Until Criterion falls below a pre-defined threshold:
  ○ Approximate $S$ with $GP$
  ○ $x_{new} \in \arg\max_{x^*} Criterion(\mu(x^*|S), \sigma^2(x^*|S))$
  ○ $S = [S\{x_{new}; f(x_{new})\}]$.
```

Box 3
The final GPO algorithm with embedded local search.

- Define initial points $\boldsymbol{x}_{init}$
- Evaluate the target function f in $\boldsymbol{x}_{init}$ and store $\boldsymbol{x}_{init}$ and $f(\boldsymbol{x}_{init})$ in the approximation set $S = [\boldsymbol{x}_{init};\boldsymbol{f}(\boldsymbol{x}_{init})]$
- Until $UCB(\mu(x^*|S),\ \sigma^2(x^*|S))$ falls below a pre-defined threshold:
    - ○ Approximate S with GP
    - ○ Create sample test $X_{sample}: \{x_i \sim N(x_{\max}, 100\sigma_\theta)\},\ i = [1 \cdot 10^6]$, where $\boldsymbol{x}_{max} = \underset{x}{\operatorname{argmax}} f(\boldsymbol{x})$
    - ○ $x_{new} = \arg\max_{x^* \in X_{sample}} UCB\left(\mu(x^*|S), \sigma^2(x^*|S)\right)$
    - ○ $S = [S\ \{x_{new};\ f(x_{new})\}\ ]$
    - ○ Start BFGS gradient descent starting at $\boldsymbol{x}_{new} \rightarrow \boldsymbol{x}_{max}$
    - ○ $S = [S\ \{x_{\max};\ f(x_{\max})\}]$.

benchmarks are the Variational Laplace (VL) method for DCM and the quasi-Newton method for HGF. Both approaches are described above and constitute fast and robust algorithms which are, however, susceptible to local minima.

A Metropolis–Hastings (MH) sampling scheme was used as an additional "ground truth" benchmark for both DCM and HGF. This method is guaranteed to find a global maximum, given an infinite number of iterations. Reliable and robust results can be obtained for large finite number of iterations, at the expense of high computational costs. This inefficiency often makes this method inapplicable in practise, but can provide a useful reference in methodological evaluations as ours.

To increase the efficiency and robustness of computations, instead of a "classical" implementation of MH we used the MCMC algorithms in *mpdcm* (Aponte et al., in preparation), a toolbox for massively parallel dynamical causal modelling which exploits the computational power of graphics processing unit (GPU). This toolbox includes an implementation of parallel tempering for DCM, an extended version of the Metropolis Hastings algorithm that simulates parallel but connected Markov Monte Carlo Chains (Swendsen and Wang, 1986; Laskey and Myers, 2003) and increases the statistical efficiency of sampling. Parallel tempering is particularly efficient when the posterior distribution is multimodal as shown by Calderhead and Girolami (2009). During the burn-in phase, the covariance of the proposal distribution of each independent chain was modified according to Shaby and Wells (2010) in order to achieve increased sampling efficiency.

For each type of analysis we ran 10 chains with 10^5 iterations each, additional to an initial burn-in period. To initialize each chain we randomly sampled from the prior. The prior variance was used as step size for each dimension. To evaluate the overall chain convergence we computed the Gelman–Rubin–Brooks multivariate potential scale reduction factor. This is one of the most widely used criteria for MCMC convergence which rests on comparing within-chain and between-chain variances of several parallel and independently initialized chains (not dissimilar to a classical ANOVA) in order to verify whether the chains have resulted in non-distinguishable distributions (for a description of both diagnostics see pp. 319–335, Brooks and Roberts, 1998).

Notably, in addition to different numerical strategies, an important difference between these optimisation schemes is the choice of objective function. That is, our GPO and MH implementations optimise the joint probability (which is proportional to the posterior), whereas Variational Laplace optimises the sufficient statistics of an approximate posterior by optimising a free-energy bound on the log evidence.

Results

Validation of numerical implementation

To demonstrate the correctness of our GPO implementation we first tested it on a well-known toy problem, using synthetic data. For this initial test we used the sinc function which has infinitely many extrema:

$$f(x) = \operatorname{sinc} kx = \frac{\sin kx}{kx}. \tag{11}$$

The larger parameter k, the more extrema the sinc function has and thus the more challenging the optimisation problem becomes. We tested three different values of $k = 0.5$, 2.0, and 5.0 with a random initialization and two kernel functions: Gaussian and a Matern 3/2 kernel. The parameters of the kernel functions were optimised using the procedure described in the "Kernel functions" subsection. Fig. 4 illustrates how well these functions are approximated by a GP with a Matern 3/2 kernel, and Table 1 summarises the convergence for the different kernel functions we used. This example demonstrates that the algorithm with a suitably chosen kernel function works fast and robust enough to be capable of solving even extremely challenging optimisation problem. In contrast to the Matern kernel, the Gaussian function tends to be overly smooth for this rapidly changing function. Note that in this case we did not include any local optimisation methods as we did not face a highly multidimensional problem.

Inversion of DCMS using synthetic data

To test the ability of GPO to address inversion problems in DCM for fMRI we generated multiple synthetic data sets from a DCM with three interacting regions (3-DCM). The structure of the model is schematically shown in Fig. 5a. Stimulus inputs include a driving input (Fig. 5b) as well as a modulatory input (Fig. 5c); these inputs mimic two types of experimental conditions. We chose to generate 128

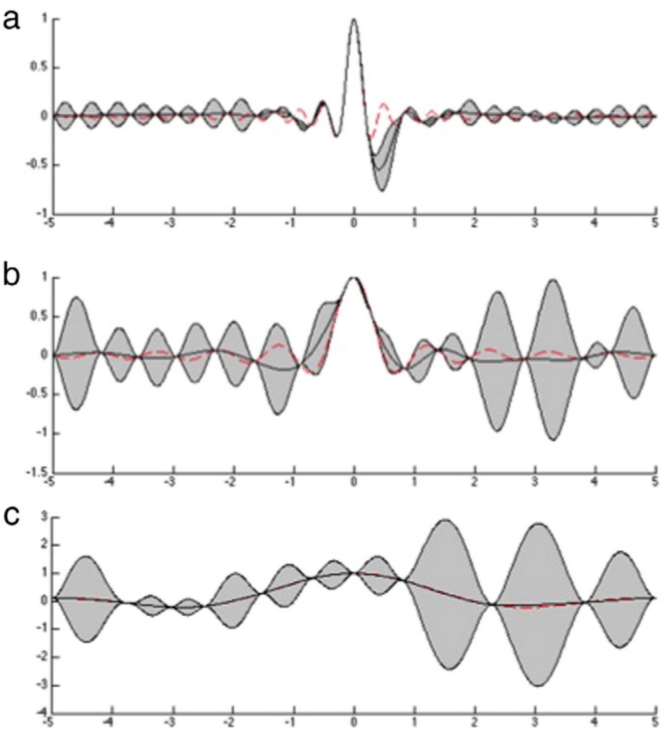

Fig. 4. Approximation of the sinc *kx* function (red dashed line) by Gaussian processes with a Matern kernel (with *k* equal to 5 (a), 2 (b) and 0.5 (c)) after 100 trials. The black line shows the mean of the estimated posterior, and grey areas reflect the variance.

Table 1
The table summarises the efficiency of GP optimisation of sinc *kx* for different values of k and two different kernels. The values reflect the average number of iterations necessary to converge to the true value.

	Squared exponential	Matern 3/2
k = 5	NA[a]	29.4 ± 14.8
k = 2	NA[a]	16.8 ± 7.7
k = 1/2	5.3 ± 1.9	8.4 ± 2.4

[a] Denotes lack of convergence after 100 iterations in more than 50% of the runs.

BOLD signal samples per region; notably, compared to most fMRI experiments this is a relatively low number of scans, particularly when considering that the model contained 20 free parameters (neuronal and haemodynamic). This small ratio between the number of data points and free parameters was a deliberate choice since we wished to examine a case which represented a fairly difficult challenge for model inversion.

For all three methods we evaluated (Variational Laplace, GPO and MH), we used the standard neuronal and haemodynamic priors from SPM8 (http://www.fil.ion.ucl.ac.uk/spm/software/spm8). All three methods used these prior means to initialize the optimisation procedure.

Our results are summarised in Fig. 6. In a first step, we chose an extremely high signal-to-noise ratio (SNR = 1000; 60 dB) to examine model inversion under conditions of very little noise. In this case, all three methods lead to very similar recovery of the known parameter values, with the Euclidian distance between true and estimated parameter values not exceeding 0.1 for any of the methods. Even in this case with very little noise, however, most estimates provided by GPO are closer to the true parameter value than the corresponding estimates by VL; it also performed slightly better than MH. This difference is reflected in the root mean squared error (RMSE) of the two methods: 0.31 for VL, 0.25 for MH and 0.15 for GP.

It should be noted that even in this case with very low noise, one would not expect any of the three methods to exactly recover the true parameter values used for data generation. This is simply because all methods optimise the posterior and not the likelihood: in Bayesian inference, the influence of the prior exerts a bias on parameter recovery whenever the prior mean does not coincide exactly with the parameter value used for data generation (as was the case here). Furthermore, even when the latter does correspond to the prior mean, parameter interdependencies induced by the likelihood function can lead to differences between posterior estimates and generating parameter values.

In a next step, we tested the performance of the different methods for high observation noise, adding as much noise as there was signal (SNR = 1; 0 dB). To ensure that our results were robust, we generated 30 different noisy time series under the same parameter values and inverted the model for each dataset separately. Fig. 7 summarises the results; error bars represent standard deviation of the mean. First, from the estimated standard deviations in Fig. 7 we can see that connectivity parameter values (except for the 4th parameter) are recovered in a fairly stable way by all three methods. Second, in this high-noise scenario, Fig. 7 shows a clear effect of the prior, leading to shrinkage of parameter estimates towards zero. Finally, we can observe that GPO is slightly more precise than the conventional VL approach in SPM. Table 2 summarises some quantitative comparisons, presenting RSME of parameter estimation as well as log joint values. A one-sample *t*-test applied to the differences between GPO and VL estimates (obtained under the same model and data) indicates that RMSE is significantly smaller for GPO ($t_{29} = 5.12$; $p < 0.05$), although the difference is not large (about 15% decrease in RMSE).

In this high-noise case, MH outperformed both GP and VL (Table 2). To ensure that this result was not affected by possible lack of convergence of MH, we computed the Gelman–Rubin–Brooks multivariate potential scale reduction factor (PSRF), where values below 1.1 indicate convergence. For the noise-free case, the PSRF was 1.0454; for the high noise case, the average PSRF value was 1.083 ± 0.03. These results indicate a reliable convergence of the MCMC algorithm.

In an additional analysis, we investigated the robustness of GPO estimates when altering the starting point of the optimization procedure and hence the ensuing sampling points. To this end, we re-estimated the parameters of one DCM ten times, under high observation noise (SNR = 1). As we initialize the algorithm stochastically, starting points varied across estimations. To assess the impact of this change, we computed the variance (across the ten model inversions) for each parameter estimate. For all parameters, all 10 estimates appeared to be almost identical, with the maximum variance being less than 10^{-4}.

We also compared the run-time of the different methods. Since this comparison depends on individual computer hardware and the specific implementation of the algorithms, the numbers should be interpreted

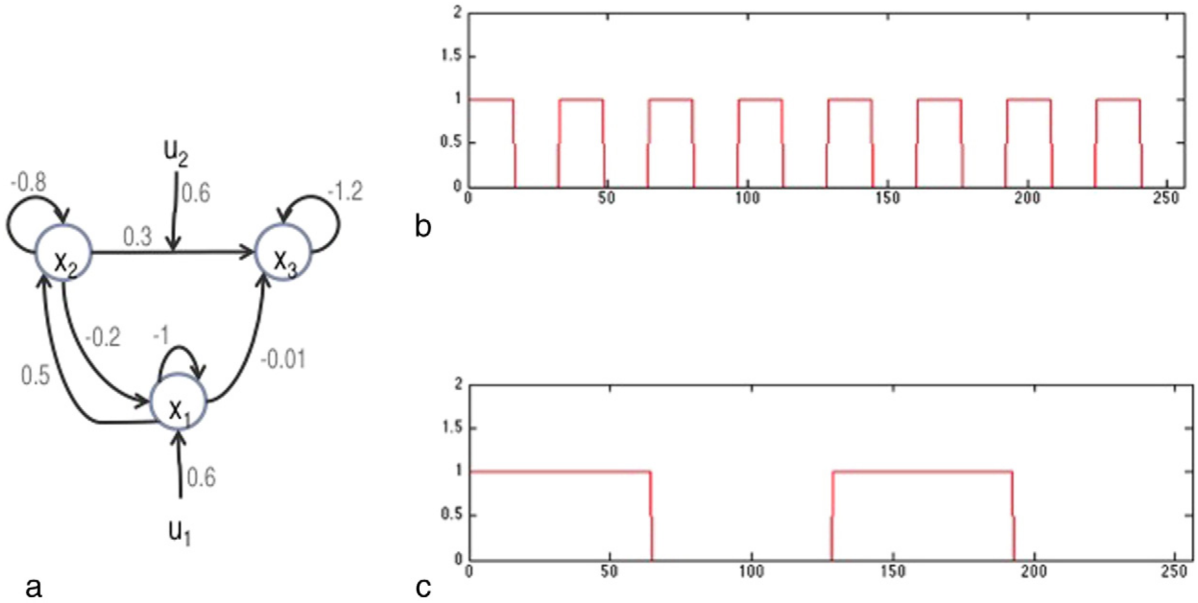

Fig. 5. The DCM which was used for simulating fMRI data. a — overall model structure, b — trajectory of driving input u_1, c — trajectory of modulatory input u_2. Small numbers in grey next to the model graph show the actual parameter values which were used to generate synthetic data.

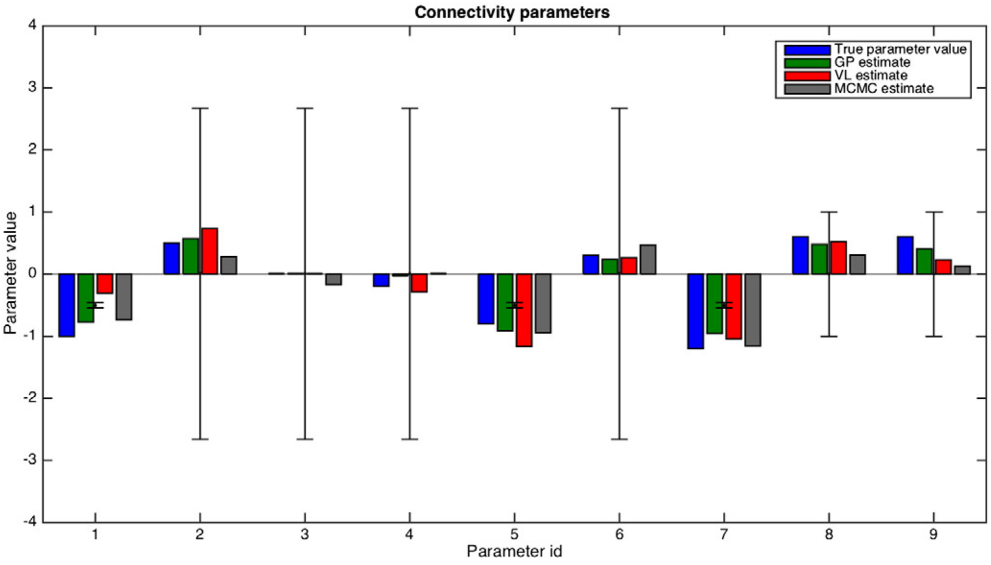

Fig. 6. Inversion of DCMs for synthetic data with virtually absent observation noise (SNR = 1000). This figure shows estimates of the neuronal parameters for the model displayed in Fig. 5: true values of the parameters (blue bars), Gaussian priors (grey bars indicate variances) and parameter estimates by VL (red bars), GP (green bars) and MCMC (black bars).

in a qualitative way and are only meant to provide a rough impression of the relative computational costs. Here, we used a Macbook Air laptop with a 1.7 GHz Intel Core i7 processor and 8 GB memory. The results for DCM inversion are summarised in Table 3: while GP optimisation was only slightly slower than VL, MH was one order of magnitude slower. Notably, this relatively good efficiency of MH was achieved by using the *mpdcm* toolbox which exploits the computational power of GPUs. For comparison, a conventional implementation of MH, which we tried initially, was two orders of magnitude slower than GP.

Analyses of empirical behavioural data using the HGF

We applied all three optimisation methods (QN, GPO and MH) to behavioural data of 16 subjects, testing 9 different variations of a three-level HGF (these models are from Diaconescu et al., 2014). For MH, we applied the same convergence criteria as in the DCM analyses above (here, the average PSRF across subjects was 1.08). The main equations of the forward model in the HGF are presented in Appendix A of this paper; for details on model inversion, please consult Mathys et al. (2011) and Diaconescu et al. (2014).

Three main mechanisms of decision making were tested: in the first three models (models 1 to 3) it was assumed that participants both tracked the volatility on the third level of HGF and incorporated the volatility in their belief-to-choice mappings; the second group of models (models 4 to 6) assumed that participants' decisions reflected their beliefs but were affected by a fixed amount of decision noise; finally, the third and simplest group (models 7 to 9), assumed that participants did not track volatility. Additionally, as subjects could base their decision on two separate sources of information (visual cue and human advice), three different response models were considered: incorporating both sources or only one of them. The first three of these models had a more complex (nonlinear) output equation than the remaining models, and model 1 was considered most likely a priori on theoretical grounds (for details, see Diaconescu et al., 2014).

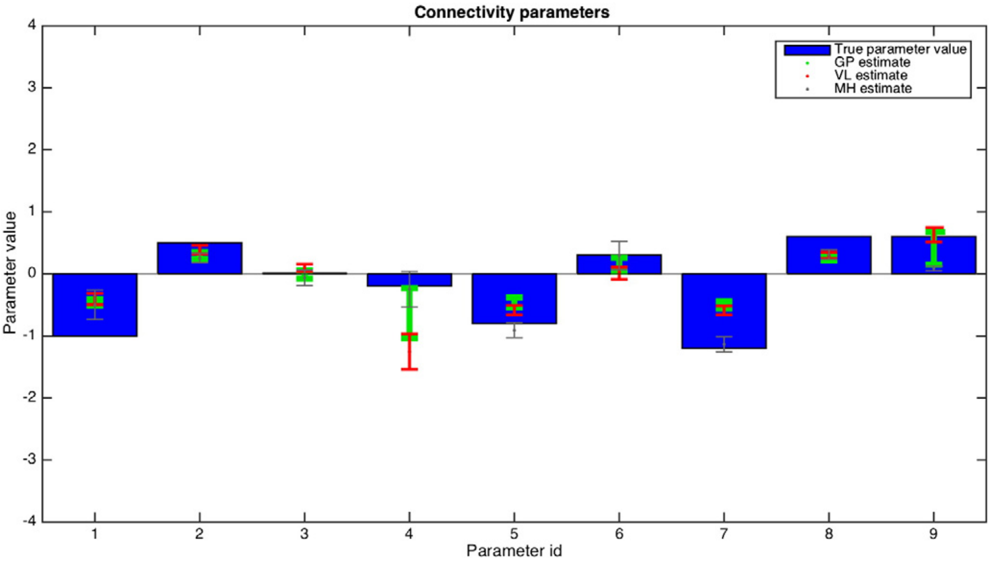

Fig. 7. Inversion of DCMs for synthetic data with high observation noise (SNR = 1). This figure shows estimates of the neuronal parameters for the model displayed in Fig. 5: true values of the parameters (blue bars), priors (grey error bars) and average parameter estimates across 30 synthetic data sets by VL (red bars), GP (green bars) and MCMC (black bars). The error bars of parameter estimates represent standard deviation of the mean.

Table 2
This table summarises the comparison between GP, VL and MH in terms of parameters RMSE and relative log joint value (divided by the log joint of true parameters) based on estimates for 30 models applied to high-noise data. RMSE is significantly smaller for GP compared to VL ($p < 0.001$) and for MH compared to VL ($p < 0.001$).

Method	GP	VL	MH
RMSE (parameter estimates)	0.42 ± 0.05	0.49 ± 0.07	0.32 ± 0.05
Relative log joint value	1.02 ± 0.01	1.01 ± 0.01	0.97 ± 0.01

Table 3
The table shows the approximate time (in minutes) required by the three methods for inverting a single DCM of the type shown in Fig. 5. Clearly, these numbers may change for different computer hardware and different implementations of the algorithms, therefore this table is only meant to demonstrate the approximate relative computational costs.

Method	Approximate analysis time cost (mins)
VL	~1.9
GP	~7.6
MH	~73.3

First, we compared our results in terms of log joint values (Fig. 8) and found that while for most models the three methods gave very similar results, for the first three models there was a notable difference between the performance of GPO and MH in comparison to QN. This behaviour signals the likely presence of local minima in the objective function. The more complex models, like model 1, appeared to be most vulnerable to QN getting stuck in local minima (Fig. 9). Note that since both GP optimisation and MH are of a stochastic nature, we repeated the analysis 50 times and provide means and standard deviations across runs in Fig. 9.

To evaluate the impact of optimisation procedure on model selection results, we computed the free energy for each model in each subject by applying a Laplace approximation to the log joint. We then performed random effects Bayesian model selection (Stephan et al., 2009) and found that while all three methods produced the same ranking of models, GPO leads to a cleaner separation of models, compared to QN (Fig. 10).

As a second validation step, we used the model parameter estimates to predict the subject's accuracy on the social learning task as well as their score from an independent questionnaire (Interpersonal Reactivity Index, IRI) which probes subjective traits of perspective-taking, a skill of central relevance for this task (the values for QN were previously reported by Diaconescu et al., 2014). We performed standard regression analysis along with Variational Bayes Regression (using the open source toolbox TAPAS: www.translationalneuromodeling.org/tapas). Checking the negative free energy as a criterion of model goodness for the GP- and MH-based regression analyses, we decided to use the estimates of all free HGF parameters for the regression analysis of task accuracy, and all free parameters except for one (κ) for the regression analysis of IRI.

For both regression methods and both prediction problems, we found that prediction performance (in terms of R^2 and negative free energy, respectively) was improved when using estimates based on GP optimisation (Tables 4, 5); the only exception was a larger R^2 for predicting task accuracy based on MH estimates. Importantly, the difference in the negative free energy (an approximation to the log evidence) between GP and the conventional QN method was larger than 3 in both cases; this value corresponds to an approximate Bayes factor of 20 and is usually considered as a threshold for decision in Bayesian model comparison (Kass & Raftery, 1995). Notably, these analyses only test for linear relations between model parameter estimates and behavioural/questionnaire data, and we cannot exclude that the three methods might perform differently if non-linear relations were considered.

Preliminary comparisons of run time are shown in Table 6, similar to DCM above. GP required, on average, a run time of the same order of magnitude as QN, not more than twice longer, while being significantly faster than MH.

Discussion

In this article, we have evaluated the utility of Gaussian processes as an alternative to MCMC and VB for inverting hierarchical Bayesian

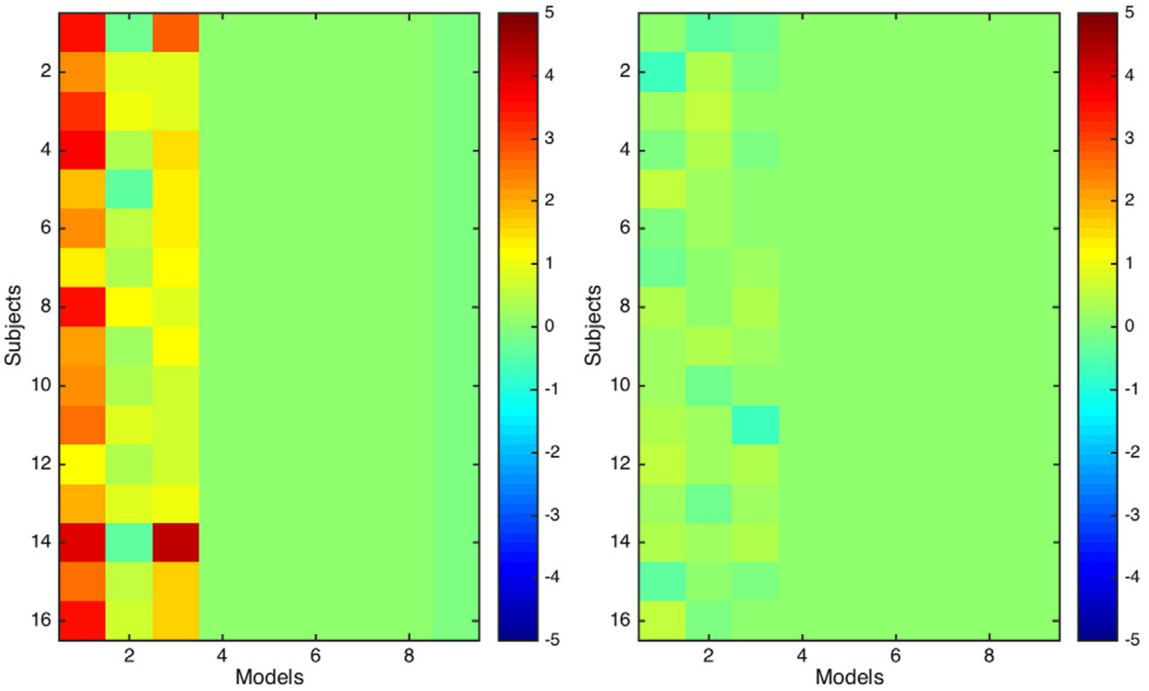

Fig. 8. Comparison of difference of log joint values across the subject (rows) and models (columns). Left panel: GP log joint–QN log joint; right panel: GP log joint–MH log joint. Green = no difference between methods, red = superiority of GP, blue = superiority of QN and MH, respectively (see colour scale).

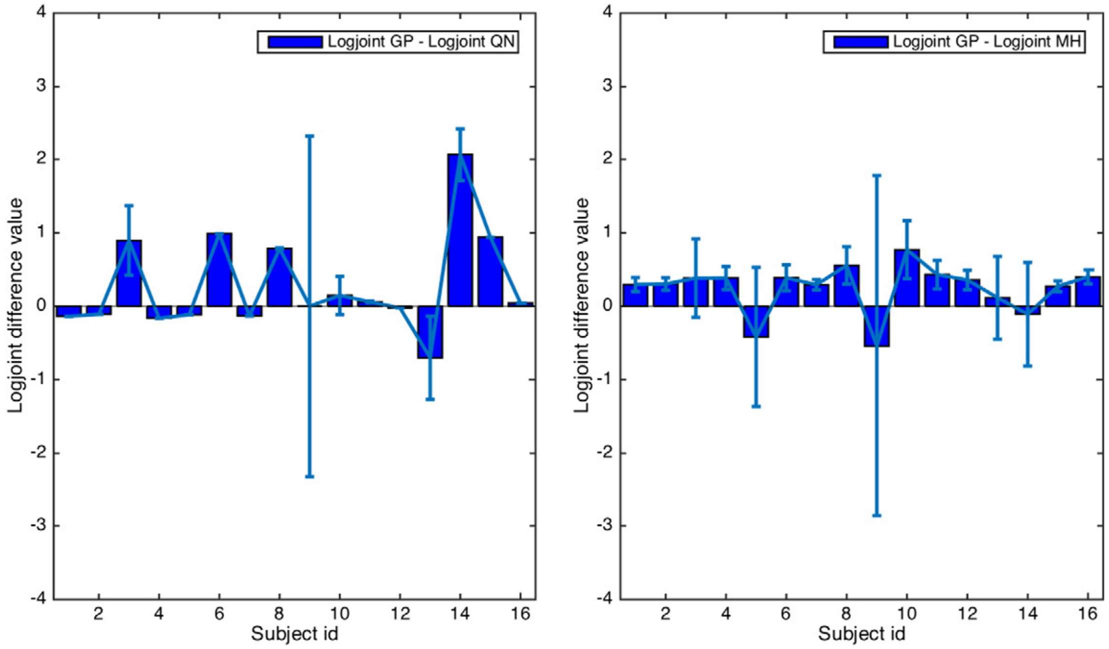

Fig. 9. Log joint difference between GP and QN (left) and GP and MH (right) for model 1 across 16 subjects, based on 50 runs of GP and MH. Error bars represent standard deviation across subjects.

models of neuroimaging and behavioural data. As a global optimisation method for sufficiently smooth objective functions, GPO is potentially less vulnerable to local extrema than VB while promising a marked increase in speed compared to MCMC. An important challenge is, however, to ensure computational efficiency when facing high-dimensional problems as they are encountered, for example, in DCM where one commonly deals with dozens of model parameters.

To address this issue, this paper proposes a variant of GPO which embeds a local search based on a quasi-Newton gradient descent. The practical utility of this implementation for inverting hierarchical Bayesian models commonly used in neuroimaging, i.e. DCM and HGF, was evaluated using both synthetic and empirical data, and the performance of GPO was benchmarked against standard methods (MCMC and VB). In this study, model inversion based on GPO yielded parameter estimates with comparable or superior accuracy to the other techniques, while being one order of magnitude faster than a highly efficient GPU-based implementation of MCMC for DCM (and two orders of magnitude compared to a conventional MCMC implementation used in the case of

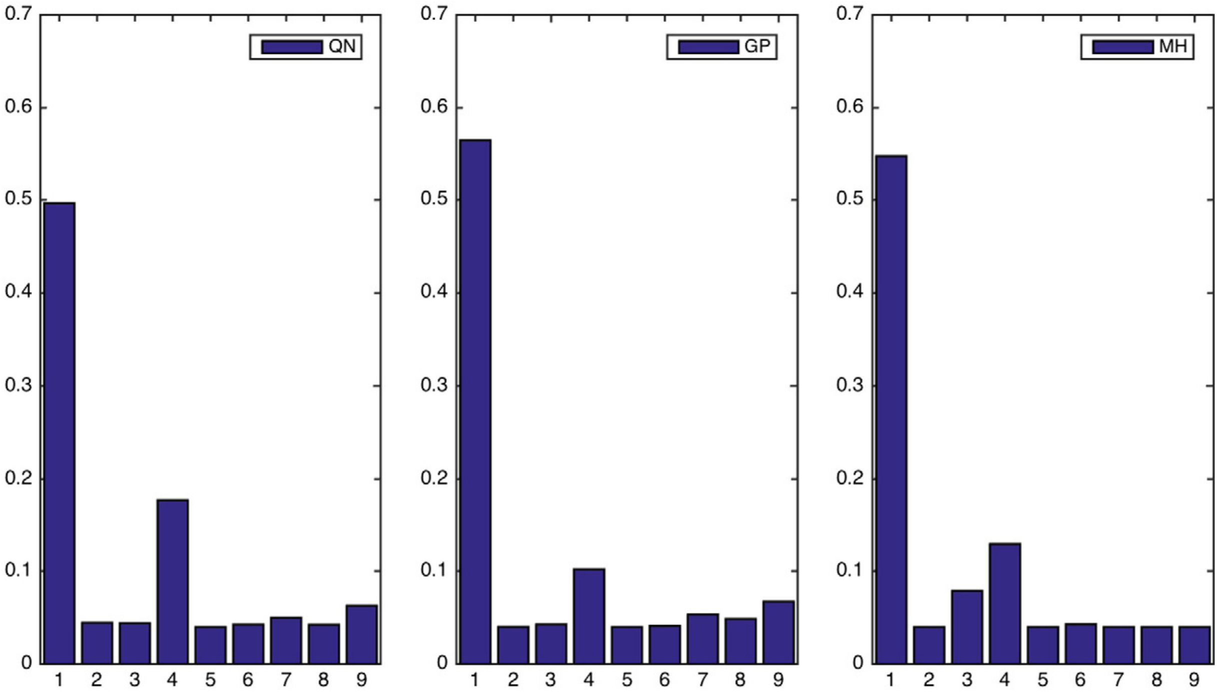

Fig. 10. Expected posterior probability of each model, given the negative free energy estimates based on all three methods (QN, GP and MH).

Table 4
The table shows R^2 for multiple linear regression and negative free energy for Variational Bayesian Regression when all model parameter estimates are used to predict the subjects' accuracy on the social learning task. A higher (more positive) negative free energy indicates a better model.

	R^2	Negative free energy
GP	0.61 ($p = 0.04$)	−134.38
QN	0.50 ($p = 0.11$)	−137.39
MH	0.62 ($p = 0.03$)	−134.54

Table 6
The table shows the approximate time (in minutes) required by the three methods for HGF model inversion. Clearly, these numbers may change for different computer hardware and different implementations of the algorithms, therefore this table is only meant to demonstrate the approximate relative computational costs.

Method	Approximate analysis time cost (mins)
QN	~1.4
GP	~8
MH	~106

the HGF). Generally, improvements in cloud computing (Armbrust et al., 2010) and the use of GPUs (Wang et al., 2013) may turn MCMC into a competitive alternative for practical use in the future. In terms of accuracy, in our high-noise simulation scenario MCMC was slightly more accurate than GPO for parameter estimation (Fig. 7, Table 2).

While a few previous studies have explored the use of Gaussian processes for identification of dynamic systems and hierarchical models (e.g., Ažman and Kocijan, 2011; Wang et al., 2008), the present work is, to our knowledge, novel in four ways. First, we present a simple but effective strategy for boosting computational performance of GPO by embedding a local gradient-based search; second, it is the first application of GPO to hierarchical Bayesian models commonly applied in neuroimaging; third, we compare the accuracy and efficiency of GPO to two competing methods (VL, MCMC); and fourth, we provide independent validation analyses for two separate datasets (one synthetic, one empirical) and models (DCM, HGF).

Our validation analyses rested on two separate approaches. For DCM, where the curse of dimensionality is more pronounced than for HGF, we generated 30 synthetic datasets with added observation noise ($db = 0$) and then challenged the different inversion methods to recover the known parameter values. This complements previous analyses of empirical fMRI data which compared VB and MCMC for inversion of DCMs (Chumbley et al., 2007). By contrast, for the HGF, simulation studies of model inversion already exist (albeit based on a simpler HGF than the one used here; Mathys et al., 2014) and we turned to empirical data. Clearly, here the "true" values of the parameters are not known, and validation has to be sought in relation to external criteria. In this case, we examined validity with respect to two criteria, i.e., which of the inversion schemes would (i) lead to cleaner discriminability of alternative models considered (in terms of [approximated] log evidence), and (ii) provide parameter estimates that better predicted an independent variable (a questionnaire score). Notably, GPO outperformed the competing methods with respect to both criteria.

The practical benefits of GPO may be of particular relevance to DCM: here, the standard estimation scheme combines the Laplace approximation with VB, rendering model inversion fast but potentially vulnerable to local extrema (for a discussion of this issue, see Daunizeau et al., 2011). A previous study (Chumbley et al., 2007) verified the robustness of this scheme for empirical fMRI data and in comparison to a Metropolis–Hastings sampling algorithm. However, this study was restricted to bilinear DCM where nonlinearities are relatively mild and restricted to the haemodynamic equations. It is conceivable for local extrema to become a more serious problem when inverting DCMs with less smoothness and more pronounced non-convexity, such as DCMs for electrophysiological data (David et al., 2006; Kiebel et al., 2007; Chen et al., 2008; Moran et al., 2009; Marreiros et al., 2010). In future work, we will optimise our present GPO implementation for this particular application domain and examine the benefits of GPO for inverting electrophysiological DCMs.

Table 5
The table shows R^2 for multivariate linear regression and negative free energy for Variational Bayesian Regression when all model parameter estimates (except for κ) are used to predict subjects' scores on an independent questionnaire (IRI). A higher (more positive) negative free energy indicates a better model.

	R^2	Negative free energy
GP	0.51 ($p = 0.04$)	−46.79
QN	0.50 ($p = 0.05$)	−50.77
MH	0.49 ($p = 0.05$)	−52.54

Future work will also extend the use of GPO from finding the mode of the posterior to approximating the posterior density itself. This discriminative approach would allow us to explicitly evaluate the model evidence (marginal likelihood) and might improve the accuracy of Bayesian model comparison.

In summary, our findings suggest that GPO is a promising alternative to established inversion schemes for hierarchical Bayesian models in neuroimaging. As a global optimisation scheme with high accuracy but acceptable computational costs even for fairly high-dimensional models, GPO may become a versatile tool for computational neuroimaging.

Software note

A MATLAB implementation of the GPO approach described in this paper will be made available as an open source code in the next release of the TAPAS Toolbox (www.translationalneuromodeling.org/tapas).

Acknowledgments

We would like to thank M. A. Osborne for helpful discussions and sharing of code, and S. Raman for helpful discussions. The authors gratefully acknowledge support by the University of Zurich (Forschungskredit to AD, core funding to KES) and by the René and Susanne Braginsky Foundation (KES).

Appendix A

This appendix provides a brief summary of the key equations of the Hierarchical Gaussian Filter (HGF). A detailed account of the model and associated update equations can be found in Mathys et al. (2011). The specific observation models used here can be found in Diaconescu et al. (2014).

The analyses in this paper refer to a three-level HGF which rests on three hierarchically coupled Gaussian random walks as state equations, where the coupling is determined by subject-specific parameters and the lowest level is linked to measured behavioural data through an observation model (Mathys et al., 2011). The lowest (first) level represents a sequence of environmental events x_1 (e.g., a binary sensory input), the second level represents a probabilistic association between environmental events (e.g., a cue-outcome contingency) x_2, and the third level encodes the log-volatility of the environment x_3. The hidden state of each level is assumed to evolve as a Gaussian random walk; critically, the variance or step size of this Gaussian random walk depends on the state at the next higher level:

$$p(x_1|x_2) = s(x)^{x_1}(1-s(x_2))^{1-x_1} = \mathrm{Bernoulli}(x_1; s(x_2)) \tag{A.1}$$

$$p\left(x_2^{(k)}\middle|x_2^{(k-1)}, x_3^{(k)}\right) = \mathcal{N}\left(x_2^{(k)}; x_2^{(k-1)}, \exp\left(\kappa x_3^{(k)} \omega\right)\right) \tag{A.2}$$

$$p\left(x_3^{(k)}\middle|x_3^{(k-1)}, \vartheta\right) = \mathcal{N}\left(x_3^{(k)}; x_3^{(k-1)}, \vartheta\right). \tag{A.3}$$

Here, k is a trial index; κ, ω, ϑ are subject-specific parameters; and s is a sigmoid function:

$$s(x) = \frac{1}{1+\exp(-x)}. \tag{A.4}$$

Applying a variational approximation to ideal hierarchical Bayesian learning under the above equations, one can derive analytical update equations (for details, see Mathys et al., 2011). When coupled to an observation model (representing a belief–choice mapping), these allow one to predict trial-wise behavioural responses. A widely used observation model is the softmax function:

$$p(y|b) = \frac{b^{\beta}}{b^{\beta}+(1-b)^{\beta}}. \tag{A.5}$$

Here, y refers to the behavioural response (e.g., choice or decision), b represents a subjectve belief (e.g., the posterior mean of x_1), and β represents decision noise. In the present analyses, models 1–3 defined β as a function of environmental log-volatility x_3 (which, in turn, evolves under nonlinear update equations; see Mathys et al., 2011) and thus treated it as a dynamic quantity. By contrast, models 4–9 defined β as a fixed parameter (models 4–9). For details, please see Diaconescu et al., 2014.

References

Aponte, E., Raman, S.S., Sengupta, B., Penny, W.D., Stephan, K.E., Heinzle, J., 2015n. mpdcm: A toolbox for Massively Parallel Dynamic Causal Modeling (in preparation).

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M., 2010. A view of cloud computing. Commun. ACM 53 (4), 50–58.

Ažman, K., Kocijan, J., 2011. Dynamical systems identification using Gaussian process models with incorporated local models. Eng. Appl. Artif. Intell. 24 (2), 398–408.

Behrens, T.E., Woolrich, M.W., Walton, M.E., Rushworth, M.F., 2007. Learning the value of information in an uncertain world. Nat. Neurosci. 10 (9), 1214–1221.

Bellman, R.E., 1957. Dynamic Programming. Princeton University Press.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer-Verlag, New York.

Brooks, S.P., Roberts, G.O., 1998. Assessing convergence of Markov chain Monte Carlo algorithms. Stat. Comput. 8, 319–335.

Calderhead, B., Girolami, M., 2009. Estimating Bayes factors via thermodynamic integration and population MCMC. Comput. Stat. Data Anal. 53 (12), 4028–4045.

Chen, C.C., Kiebel, S.J., Friston, K.J., 2008. Dynamic causal modelling of induced responses. NeuroImage 41 (4), 1293–1312.

Chumbley, J.R., Friston, K.J., Fearn, T., Kiebel, S.J., 2007. A Metropolis–Hastings algorithm for dynamic causal models. NeuroImage 38 (3), 478–487.

D'Ardenne, K., McClure, S.M., Nystrom, L.E., Cohen, J.D., 2008. BOLD responses reflecting dopaminergic signals in the human ventral tegmental area. Science 319 (5867), 1264–1267.

Daunizeau, J., David, O., Stephan, K.E., 2011. Dynamic causal modelling: a critical review of the biophysical and statistical foundations. NeuroImage 58 (2), 312–322.

Daunizeau, J., Adam, V., Rigoux, L., 2014. VBA: a probabilistic treatment of nonlinear models for neurobiological and behavioural data. PLoS Comput. Biol. 10 (1), e1003441.

David, O., Kiebel, S.J., Harrison, L.M., Mattout, J., Kilner, J.M., Friston, K.J., 2006. Dynamic causal modeling of evoked responses in EEG and MEG. NeuroImage 30 (4), 1255–1272.

Diaconescu, A.D., Mathys, C., Weber, L.A.E., Daunizeau, J., Kasper, L., Lomakina, E.I., Fehr, E., Stephan, K.E., 2014. Inferring on the intentions of others by hierarchical Bayesian learning. PLoS Comput. Biol. 10 (9), e1003810.

Duvenaud, D., 2014. Automatic Model Construction with Gaussian Processes (PhD Thesis). University of Cambridge.

Frean, M., Boyle, P., 2008. Using Gaussian processes to optimize expensive functions. Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, pp. 258–267.

Friston, K.J., Dolan, R.J., 2010. Computational and dynamic models in neuroimaging. NeuroImage 52 (3), 752–765.

Friston, K.J., Mechelli, A., Turner, R., Price, C.J., 2000. Nonlinear responses in fMRI: the balloon model, Volterra kernels, and other hemodynamics. NeuroImage 12, 466–477.

Friston, K.J., Harrison, L., Penny, W., 2003. Dynamic causal modelling. NeuroImage 19 (4), 1273–1302.

Friston, K.J., Mattout, J., Trujillo-Barreto, N., Ashburner, J., Penny, W., 2007. Variational free energy and the Laplace approximation. NeuroImage 34 (1), 220–234.

Glascher, J.P., O'Doherty, J.P., 2010. Model-based approaches to neuroimaging: combining reinforcement learning theory with fMRI data. Wiley Interdiscip. Rev. Cogn. Sci. 1 (4), 501–510.

Iglesias, S., Mathys, C., Brodersen, K.H., Kasper, L., Piccirelli, M., den Ouden, H.E., Stephan, K.E., 2013. Hierarchical prediction errors in midbrain and basal forebrain during sensory learning. Neuron 80 (2), 519–530.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. J. Glob. Optim. 13 (4), 455–492.

Kass, R.E., Raftery, A.E., 1995. Bayes factors. J. Am. Stat. Assoc. 90 (430), 773–795.

Kiebel, S.J., Garrido, M.I., Friston, K.J., 2007. Dynamic causal modelling of evoked responses: the role of intrinsic connections. NeuroImage 36 (2), 332–345.

Krause, A., Singh, A., Guestrin, C., 2008. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies. J. Mach. Learn. Res. 9, 235–284.

Krige, D., 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. J. Chem. Metall. Min. Soc. S. Afr. 52 (6), 119–139.

Laskey, K.B., Myers, J.W., 2003. Population Markov chain Monte Carlo. Mach. Learn. 50 (1–2), 175–196.

MacKay, D.J.C., 2003. Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge.

Marquand, A., Howard, M., Brammer, M., Chu, C., Coen, S., Mourão-Miranda, J., 2010. Quantitative prediction of subjective pain intensity from whole-brain fMRI data using Gaussian processes. NeuroImage 49 (3), 2178–2189. http://dx.doi.org/10.1016/j.neuroimage.2009.10.072.

Marreiros, A.C., Kiebel, S.J., Friston, K.J., 2010. A dynamic causal model study of neuronal population dynamics. NeuroImage 51 (1), 91–101.

Mathys, C., Daunizeau, J., Friston, K.J., Stephan, K.E., 2011. A Bayesian foundation for individual learning under uncertainty. Front. Hum. Neurosci. 5, 39.

Mathys, C.D., Lomakina, E.I., Daunizeau, J., Iglesias, S., Brodersen, K.H., Friston, K.J., Stephan, K.E., 2014. Uncertainty in perception and the Hierarchical Gaussian Filter. Front. Hum. Neurosci. 8, 825. http://dx.doi.org/10.3389/fnhum.2014.00825.

Metropolis, N., Ulam, S., 1949. The Monte Carlo method. J. Am. Stat. Assoc. 44 (247), 335–341.

Mockus, J., Tiesis, V., Zilinskas, A., 1978. The application of Bayesian methods for seeking the extremum. Towards Glob. Optimisation 2, 117–129.

Moran, R.J., Stephan, K.E., Seidenbecher, T., Pape, H.C., Dolan, R.J., Friston, K.J., 2009. Dynamic causal models of steady-state responses. NeuroImage 44 (3), 796–811.

Mourão-Miranda, J., Oliveira, L., Ladouceur, C.D., Marquand, A., Brammer, M., Birmaher, B., Phillips, M.L., 2012. Pattern recognition and functional neuroimaging help to discriminate healthy adolescents at risk for mood disorders from low risk adolescents. PLoS One 7 (2).

Nocedal, J., Wright, S.J., 2006. Numerical Optimization. 2nd ed. Springer-Verlag, New York.

O'Doherty, J.P., Dayan, P., Friston, K., Critchley, H., Dolan, R.J., 2003. Temporal difference models and reward-related learning in the human brain. Neuron 38 (2), 329–337.

Osborne, M.A., Garnett, R., Roberts, S.J., 2009. Gaussian processes for global optimization. International Conference on Learning and Intelligent Optimization (LION 2009).

Penny, W.D., Stephan, K.E., Mechelli, A., Friston, K.J., 2004. Comparing dynamic causal models. NeuroImage 22 (3), 1157–1172.

Preuschoff, K., Quartz, S.R., Bossaerts, P., 2008. Human insula activation reflects risk prediction errors as well as risk. J. Neurosci. 28 (11), 2745–2752.

Pyka, M., Hahn, T., Heider, D., Krug, A., Sommer, J., Kircher, T., Jansen, A., 2013. Baseline activity predicts working memory load of preceding task condition. Hum. Brain Mapp. 34 (11), 3010–3022. http://dx.doi.org/10.1002/hbm.22121.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. MIT Press.

Rescorla, R.A., Wagner, A.R., 1972. A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In: Black, A.H., Prokasy, W.F. (Eds.), Classical Conditioning II: Current Research and Theory. Appleton Century Crofts, New York, pp. 64–99.

Salimi-Khorshidi, G., Nichols, T.E., Smith, S.M., Woolrich, M.W., 2011. Using Gaussian-process regression for meta-analytic neuroimaging inference based on sparse observations. IEEE Trans. Med. Imaging 30 (7), 1401–1416.

Schultz, W., Dayan, P., Montague, P.R., 1997. A neural substrate of prediction and reward. Science 275 (5306), 1593–1599.

Shaby, B., Wells, M.T., 2010. Exploring an adaptive metropolis algorithm. Technical Report.

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M., 2010. Gaussian process optimization in the bandit setting: no regret and experimental design. Proceedings of the 27th International Conference on Machine Learning.

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M., 2012. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Trans. Inf. Theory 58 (5), 3250–3265. http://dx.doi.org/10.1109/TIT.2011.2182033.

Stephan, K.E., Weiskopf, N., Drysdale, P.M., Robinson, P.A., Friston, K.J., 2007. Comparing hemodynamic models with DCM. NeuroImage 38, 387–401.

Stephan, K.E., Penny, W.D., Daunizeau, J., Moran, R.J., Friston, K.J., 2009. Bayesian model selection for group studies. NeuroImage 46, 1004–1017.

Swendsen, R., Wang, J., 1986. Replica Monte Carlo simulation of spin-glasses. Phys. Rev. Lett. 57, 2607–2609.

Vezhnevets, A., Ferrari, V., Buhmann, J., 2012. Weakly supervised structured output learning for semantic segmentation. Computer Vision and Pattern Recognition (CVPR). IEEE, Rhode Island, USA, pp. 845–852.

Wang, J.M., Fleet, D.J., Hertzmann, A., 2008. Gaussian process dynamical models for human motion. IEEE Trans. Pattern Anal. Mach. Intell. 30 (2), 283–298.

Wang, W.J., Hsieh, I.F., Chen, C.C., 2013. Accelerating computation of DCM for ERP in MATLAB by external function calls to the GPU. PLoS One 8 (6), e66599.