Qué es el análisis bayesiano de Problemas Inversos

Marcos A. Capistrán[1] J. Andrés Christen[1]
Sophie Donnet[2]

[1]Centro de Investigación en Matemáticas , CIMAT, Guanajuato, Mexico

[2]AgroParisTech, Francia.

"Seminario Aleatorio" ITAM, 19 FEB 2016

Introduction, Bayesian Analysis of Inverse Problems

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...* If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K.* Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Introduction, Bayesian Analysis of Inverse Problems

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...*
If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K.* Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Introduction, Bayesian Analysis of Inverse Problems

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...* If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K.* Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Introduction, Bayesian Analysis of Inverse Problems

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...* If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K.* Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Introduction, Bayesian Analysis of Inverse Problems

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...* If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K.* Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Introduction, Bayesian Analysis of Inverse Problems

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...* If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K*. Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Problem 1: Model the growth of bacteria in a closed laboratory environment with limited resources. *Bacteria will increase exponentially ...* If X is the number of Bacteria then

$$\frac{dX}{dt} = \lambda X(t),$$

exponential growth, Malthus model. *until an essential nutrient is exhausted, reaching a maximum K.* Perhaps then:

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{1}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$. Malthus-Verhulst growth model.

For $\theta = (\lambda, K)$, finding $X_\theta(t)$ is the **Direct Problem**.

Problem 2: Now suppose you have observations y_i on the number of bacteria a times $t_1, \ldots, t_n \in [0, T]^n$, and assuming a model for bacteria growth $X_\theta(t)$, *what can be said about the unknown parameters* θ*?*

Since now we observe (somehow) $X_\theta(t)$ and we want to know abot θ, this is the **Inverse Problem**.

Problem 2: Now suppose you have observations y_i on the number of bacteria a times $t_1, \ldots, t_n \in [0, T]^n$, and assuming a model for bacteria growth $X_\theta(t)$, *what can be said about the unknown parameters* θ*?*

Since now we observe (somehow) $X_\theta(t)$ and we want to know abot θ, this is the **Inverse Problem**.

Problem 2: Now suppose you have observations y_i on the number of bacteria a times $t_1, \ldots, t_n \in [0, T]^n$, and assuming a model for bacteria growth $X_\theta(t)$, *what can be said about the unknown parameters* θ*?*

Since now we observe (somehow) $X_\theta(t)$ and we want to know abot θ, this is the **Inverse Problem**.

In General:

Assume that we observe a process $\mathbf{y} = (y_1, \ldots, y_n)$ at the discrete times $t_1, \ldots, t_n \in [0, T]^n$ such that

$$y_i = f(X_\theta(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2) \qquad (\mathcal{M}). \tag{2}$$

where X_θ is the solution of the following system of ordinary differential equations, namely the the regressor or the **Forward Model**,

$$\frac{dX_\theta}{dt} = F(X_\theta, t, \theta); \quad X_\theta(t_0) = X_0. \tag{3}$$

$\theta \in \Theta \subset \mathbb{R}^d$ is a vector of unknown parameters.
$F : \mathbb{R}^p \times [0, T] \times \Theta \mapsto \mathbb{R}^p$ is a known function[1].

From the data $\mathbf{y}$, now we want to *know* about the parameters θ. This is the **Inverse Problem.**

In General:
Assume that we observe a process $\mathbf{y} = (y_1, \ldots, y_n)$ at the discrete times $t_1, \ldots, t_n \in [0, T]^n$ such that

$$y_i = f(X_\theta(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2) \qquad (\mathcal{M}). \tag{2}$$

where X_θ is the solution of the following system of ordinary differential equations, namely the the regressor or the **Forward Model**,

$$\frac{dX_\theta}{dt} = F(X_\theta, t, \theta); \quad X_\theta(t_0) = X_0. \tag{3}$$

$\theta \in \Theta \subset \mathbb{R}^d$ is a vector of unknown parameters.
$F : \mathbb{R}^p \times [0, T] \times \Theta \mapsto \mathbb{R}^p$ is a known function[1].

From the data $\mathbf{y}$, now we want to *know* about the parameters θ. This is the **Inverse Problem.**

Introduction, Bayesian Analysis of Inverse Problems

We may regard this problem as a mapping

$$\mathcal{F}_{\mathbf{t}}(\theta) = (f(X_\theta(t_1)), \ldots, f(X_\theta(t_n))),$$

this is the forward map. The inverse mapping is in general ill posed, and does not make much sense:

$$\text{“}\mathcal{F}_{\mathbf{t}}^{-1}(y_1, \ldots, y_n) = \theta''.$$

Something else needs to be done/assumed, like a “regularization” strategy, a noise model etc. or

Uncertainty Quantification (UQ) using bayesian inference.

We have then

$$y_i = f(X_\theta(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2). \tag{4}$$

Let, **for a second** that $X_\theta(t) = \theta_0 + \theta_1 t + \theta_2 t^2$ and $f(x) = x$, a linear model, or $f(x) = e^x$, a Generalized Linear Model, etc. This is a usual statistical problem. This is done everyday in statistics!

$X_\theta(t_i)$ is some regressor with parameters θ, $f(x)$ is a link function, y_i is data and the model is additive Gaussian independent noise.

We have then

$$y_i = f(X_\theta(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2). \tag{4}$$

Let, **for a second** that $X_\theta(t) = \theta_0 + \theta_1 t + \theta_2 t^2$ and $f(x) = x$, a linear model, or $f(x) = e^x$, a Generalized Linear Model, etc. This is a usual statistical problem. This is done everyday in statistics!

$X_\theta(t_i)$ is some regressor with parameters θ, $f(x)$ is a link function, y_i is data and the model is additive Gaussian independent noise.

We have then

$$y_i = f(X_\theta(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2). \tag{4}$$

Let, **for a second** that $X_\theta(t) = \theta_0 + \theta_1 t + \theta_2 t^2$ and $f(x) = x$, a linear model, or $f(x) = e^x$, a Generalized Linear Model, etc. This is a usual statistical problem. This is done everyday in statistics!

$X_\theta(t_i)$ is some regressor with parameters θ, $f(x)$ is a link function, y_i is data and the model is additive Gaussian independent noise.

Introduction, Bayesian Analysis of Inverse Problems

In general the joint distribution for data is

$$f(\mathbf{y}|\theta, \sigma)$$

where $E(y_i|\theta, \sigma) = f(X_\theta(t_i))$, and by observing the data $\mathbf{y}$ we want to *infer* θ ...

The Inverse Problem *is* an inference problem.

Introduction, Bayesian Analysis of Inverse Problems

In general the joint distribution for data is

$$f(\mathbf{y}|\theta, \sigma)$$

where $E(y_i|\theta, \sigma) = f(X_\theta(t_i))$, and by observing the data $\mathbf{y}$ we want to *infer* θ ...

The Inverse Problem *is* an inference problem.

Introduction, Bayesian Analysis of Inverse Problems

In general the joint distribution for data is

$$f(\mathbf{y}|\theta, \sigma)$$

where $E(y_i|\theta, \sigma) = f(X_\theta(t_i))$, and by observing the data $\mathbf{y}$ we want to *infer* θ ...

The Inverse Problem *is* an inference problem.

The Bayesian approach for inference

Uncertainty (formally defined) is **Quantified** (UQ) with a probability measure. The agent interested in knowing about θ, establishes a random variable Θ with its probability density

$$P_{\Theta}(\cdot).$$

The values Θ takes are the possible values for the parameters.

This probability measure quantifies the uncertainty the agent has regarding the possible values for the parameters in the model.

$P_{\Theta}(\theta)$ is called the *a priori* distribution.

The Bayesian approach for inference

In the presence of data $\mathbf{Y} = \mathbf{y}$, and assuming a model for $\mathbf{Y}$ $P_{\mathbf{Y}|\Theta}(\mathbf{y}|\theta)$, the Bayesian theory prescribes to

calculate the conditional distribution of the unknowns given the data. That is

$$P_{\Theta|\mathbf{Y}}(\theta|\mathbf{y}) = \frac{P_{\mathbf{Y}|\Theta}(\mathbf{y}|\theta)P_{\Theta}(\theta)}{P_{\mathbf{Y}}(\mathbf{y})},$$

which is conveniently calculated using Bayes' theorem (and thus its name!).

$P_{\Theta|\mathbf{Y}}(\theta|\mathbf{y})$ is called the *a posteriori* distribution.

Bayesian Inference for Inverse Problems

In the above context, we include the observational noise as an unknown, therefore the *thoeretical* posterior distribution is

$$P_{\mathbf{\Phi}|\mathbf{Y}}(\theta,\sigma|\mathbf{y}) = \frac{P_{\mathbf{Y}|\mathbf{\Phi}}(\mathbf{y}|\theta,\sigma)P_{\mathbf{\Phi}}(\theta,\sigma)}{P_{\mathbf{Y}}(\mathbf{y})}. \tag{5}$$

where $P_{\mathbf{\Phi}}(\theta,\sigma)$ is the prior distribution on $\mathbf{\Phi} = (\theta,\sigma)$ and

$$P_{\mathbf{Y}}(\mathbf{y}) = \int P_{\mathbf{Y}|\mathbf{\Phi}}(\mathbf{y}|\theta,\sigma)P_{\mathbf{\Phi}}(\theta,\sigma)d\theta d\sigma$$

is the normalization constant, also called the **marginal likelihood** of data $\mathbf{y}$.

Bayesian Inference for Inverse Problems

In the above context, we include the observational noise as an unknown, therefore the *thoeretical* posterior distribution is

$$P_{\mathbf{\Phi}|\mathbf{Y}}(\theta,\sigma|\mathbf{y}) = \frac{P_{\mathbf{Y}|\mathbf{\Phi}}(\mathbf{y}|\theta,\sigma)P_{\mathbf{\Phi}}(\theta,\sigma)}{P_{\mathbf{Y}}(\mathbf{y})}. \tag{5}$$

where $P_{\mathbf{\Phi}}(\theta,\sigma)$ is the prior distribution on $\mathbf{\Phi} = (\theta,\sigma)$ and

$$P_{\mathbf{Y}}(\mathbf{y}) = \int P_{\mathbf{Y}|\mathbf{\Phi}}(\mathbf{y}|\theta,\sigma)P_{\mathbf{\Phi}}(\theta,\sigma)d\theta d\sigma$$

is the normalization constant, also called the **marginal likelihood** of data $\mathbf{y}$.

Bayesian Inference for Inverse Problems

Therefore, the likelihood function

$$P_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma) = \sigma^{-n}(2\pi)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f(X_\theta(t_i)))^2\right\} \qquad (6)$$

where $\Phi = (\Theta, \Sigma)$ is a random variable with particular realizations $\phi = (\theta, \sigma)$.

This expression involves the computation of X_θ, a solution of the ODE!. However, except in very simple cases, an explicit expression of the solution is in general not available (although its existence is ensured by the regularity conditions on F; the lhs of the ODE system $\frac{dX}{dt} = F(X, \theta)$.)

Bayesian Inference for Inverse Problems

As a consequence, in practice, the ODE system is solved using a numerical solver and inference is performed, not on the previous "exact" model but on an approximate model, namely

$$y_i = f(X_\theta^h(t_i)) + \varepsilon_i, \quad \varepsilon_i \sim_{i.i.d.} \mathcal{N}(0, \sigma^2) \qquad (\mathcal{M}_h) \qquad (7)$$

where X_θ^h denotes the approximate solution of (3) supplied by the numerical solver (h being a precision parameter of the solver, typically its step size). The new likelihood derived from model $\mathcal{M}_h$ is thus

$$P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta, \sigma) = \sigma^{-n}(2\pi)^{-n/2} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(X_\theta^h(t_i)))^2 \right\}.$$

ODE solvers and Bayes

Therefore, in general, we need to use a numerical method to approximate $X_\theta(t)$ with $X_\theta^h(t)$ for some precision h.

Linear Multistep Methods, like the Adams–Bashforth or Adams–Moulton methods for non-stiff ODE's are preferred in the literature (or the backward differentiation formulae (BDF) for stiff systems are favored in the literature).

In our experience we have worked with the LSODE FORTRAN package, which is now available in a series of platforms including Python-SciPy and R, that dynamically choose between precisely the former mentioned solvers.

ODE solvers and Bayes

Therefore, in general, we need to use a numerical method to approximate $X_\theta(t)$ with $X_\theta^h(t)$ for some precision h.

Linear Multistep Methods, like the Adams–Bashforth or Adams–Moulton methods for non-stiff ODE's are preferred in the literature (or the backward differentiation formulae (BDF) for stiff systems are favored in the literature).

In our experience we have worked with the LSODE FORTRAN package, which is now available in a series of platforms including Python-SciPy and R, that dynamically choose between precisely the former mentioned solvers.

ODE solvers and Bayes

Therefore, in general, we need to use a numerical method to approximate $X_\theta(t)$ with $X_\theta^h(t)$ for some precision h.

Linear Multistep Methods, like the Adams–Bashforth or Adams–Moulton methods for non-stiff ODE's are preferred in the literature (or the backward differentiation formulae (BDF) for stiff systems are favored in the literature).

In our experience we have worked with the LSODE FORTRAN package, which is now available in a series of platforms including Python-SciPy and R, that dynamically choose between precisely the former mentioned solvers.

ODE solvers and Bayes: Global error control

As a fact, for any explicit one-step method of order p such as Euler ($p = 1$) and Runge-Kutta ($p = 2$ or $p = 4$) schemes, the global error is of order $O(h^p)$ for h small enough. That is

$$\max_{t \in \{t_1, t_2, \ldots, t_n\}} ||X_\theta(t) - X_\theta^h(t)|| \leq C_\theta h^p.$$

The Adams–Bashforth or Adams–Moulton methods have global error order of $p \geq 4$.

This global error order control is the only assumption needed to prove our main result.

Bayesian Inference for Inverse Problems

The *numerical* posterior distribution is

$$P^h_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y}) = \frac{P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)P_\Phi(\theta,\sigma)}{P^h_{\mathbf{Y}}(\mathbf{y})} \tag{8}$$

where $P_\Phi(\theta,\sigma)$ is the prior distribution on (θ,σ) and

$$P^h_{\mathbf{Y}}(\mathbf{y}) = \int P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)P_\Phi(\theta,\sigma)d\theta d\sigma$$

is the normalization constant, also called the **marginal likelihood** of data $\mathbf{y}$.
Since there is no alternative but to use the numerical posterior, there exists a real need in understanding and controlling the error made in working with $P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)$ instead of $P_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)$.

We further elaborate on this problem and present some recent work by the authors.

Bayesian Inference for Inverse Problems

The *numerical* posterior distribution is

$$P^h_{\Phi|\mathbf{Y}}(\theta,\sigma|\mathbf{y}) = \frac{P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)P_\Phi(\theta,\sigma)}{P^h_{\mathbf{Y}}(\mathbf{y})} \tag{8}$$

where $P_\Phi(\theta,\sigma)$ is the prior distribution on (θ,σ) and

$$P^h_{\mathbf{Y}}(\mathbf{y}) = \int P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)P_\Phi(\theta,\sigma)d\theta d\sigma$$

is the normalization constant, also called the **marginal likelihood** of data $\mathbf{y}$.
Since there is no alternative but to use the numerical posterior, there exists a real need in understanding and controlling the error made in working with $P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)$ instead of $P_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma)$.

We further elaborate on this problem and present some recent work by the authors.

Bayes Factors

Main idea: **Use the Bayesian 'model' comparison methodology (ie. Bayes Factors) when analyzing the numerical vs. the theoretical versions of the resulting posterior distribution.**

The Bayesian model comparison and model averaging tools, in particular pairwise model comparison using Bayes Factors, is in such case the main tool to be used in this context, as far as predictive power is concerned Hoeting *et al.* (1999).

That is, compare $P_{\Theta|Y}(\theta|y)$ vs $P^h_{\Theta|Y}(\theta|y)$ as statistical models. That is, model $\mathcal{M}$ vs. $\mathcal{M}_h$.

Approximations

Recently a series of papers, for example Schwab and Stuart (2012), discuss regularity conditions under which $P^h_{\Theta|Y}(\theta|y)$ tends to the $P_{\Theta|Y}(\theta|y)$ as the approximation error (h) tends to zero, using a suitable metric.

A metric comparison (ie. $||P_{\Theta|Y}(\cdot|y) - P^h_{\Theta|Y}(\cdot|y)||$) is useful to proving the required convergence theorems, **but more practical considerations** will be needed when evaluating the relative benefits of a numerical approach with a particular solver step size $h > 0$ (for data y).

Approximations

Note that both $P_{\Theta|Y}(\cdot|y)$ and $P^{h}_{\Theta|Y}(\cdot|y)$ may be compared as **models**,

being $P_{\Theta|Y}(\cdot|y)$ the reference and **only theoretically available** posterior and the approximate $P^{h_i}_{\Theta|Y}(\cdot|y)$, for various solver precisions $h_1 < h_2 < \cdots$, as alternative and decreasingly less computationally demanding competing models.

Bayes factors may then be used to establish a sound comparison, to balance predictive power on the one hand vs. solver CPU time on the other, to establish a useful solver precision.

Approximations

Note that both $P_{\Theta|Y}(\cdot|y)$ and $P^{h}_{\Theta|Y}(\cdot|y)$ may be compared as **models**,

being $P_{\Theta|Y}(\cdot|y)$ the reference and **only theoretically available** posterior and the approximate $P^{h_i}_{\Theta|Y}(\cdot|y)$, for various solver precisions $h_1 < h_2 < \cdots$, as alternative and decreasingly less computationally demanding competing models.

Bayes factors may then be used to establish a sound comparison, to balance predictive power on the one hand vs. solver CPU time on the other, to establish a useful solver precision.

Bayes Factors and Bayesian model selection

In Bayesian inference, model selection is performed using the Bayes factors whose principle is recalled here in a general context. Let $\mathbf{y}$ be the observations and let $\mathcal{M}$ and $\mathcal{M}_h$ the exact theoretical and the approximate numerical statistical models, arising from:

$$\mathcal{M} = \left\{ \begin{array}{lcl} \mathbf{y} & \sim & P_{\mathbf{Y}|\mathbf{\Phi}}(\mathbf{y}|\phi) \\ \phi & \sim & P_{\mathbf{\Phi}}(\phi) \end{array} \right. \qquad \mathcal{M}_h = \left\{ \begin{array}{lcl} \mathbf{y} & \sim & P^h_{\mathbf{Y}|\mathbf{\Phi}}(\mathbf{y}|\phi) \\ \phi & \sim & P_{\mathbf{\Phi}}(\phi). \end{array} \right.$$

where $\phi = (\theta, \sigma)$.

Bayesian model selection

Consider a prior distribution on the set of the models $\{\mathcal{M}, \mathcal{M}_h\}$, the decision between the competing models $\mathcal{M}$ and $\mathcal{M}_h$ is based on the ratio of their respective posterior probabilities

$$\frac{P(\mathcal{M}|\mathbf{y})}{P(\mathcal{M}_h|\mathbf{y})} = \frac{P_{\mathbf{Y}}(\mathbf{y})}{P^h_{\mathbf{Y}}(\mathbf{y})} \frac{P(\mathcal{M})}{P(\mathcal{M}_h)}$$

where $P^i_{\mathbf{Y}}(\mathbf{y})$ is the 'integrated likelihood' or the marginal distribution of $\mathbf{Y}$ of model $\mathcal{M}_i$, namely

$$P^h_{\mathbf{Y}}(\mathbf{y}) = \int P^h_{\mathbf{Y}|\Phi}(\mathbf{y}|\theta,\sigma) P_{\Phi}(\theta,\sigma) d\theta d\sigma$$

where $\mathcal{M}_0 = \mathcal{M}$. In fact, this is the normalization constant for $\mathcal{M}_h$.

MCMC

Simulation from the posterior distribution is not a direct task and Markov Chain Monte Carlo (MCMC) algorithms are standard tools to sample from the posterior distribution $P^h_{\mathbf{\Phi}|\mathbf{Y}}(\theta, \sigma|\mathbf{y})$.

For nonlinear forward maps **multimodality is very common**.

In our examples we use the t-walk Christen and Fox (2010), a self adjusted, generic MCMC for continuos distributions `http://www.cimat.mx/~jac/twalk/`. This is an affine invariant MCMC. Also, inspired by the t-walk, there is the emcee (the MCMC hammer) `http://dan.iel.fm/emcee/current/`.

Has no tuning parameters, and is suitable for mutimodal posteriors ... and is implemented in R, Python, C++, C and Matlab.

MCMC

Simulation from the posterior distribution is not a direct task and Markov Chain Monte Carlo (MCMC) algorithms are standard tools to sample from the posterior distribution $P^h_{\mathbf{\Phi}|\mathbf{Y}}(\theta, \sigma|\mathbf{y})$.

For nonlinear forward maps **multimodality is very common**.

In our examples we use the t-walk Christen and Fox (2010), a self adjusted, generic MCMC for continuos distributions `http://www.cimat.mx/~jac/twalk/`. This is an affine invariant MCMC. Also, inspired by the t-walk, there is the emcee (the MCMC hammer) `http://dan.iel.fm/emcee/current/`.

Has no tuning parameters, and is suitable for mutimodal posteriors ... and is implemented in R, Python, C++, C and Matlab.

MCMC

Simulation from the posterior distribution is not a direct task and Markov Chain Monte Carlo (MCMC) algorithms are standard tools to sample from the posterior distribution $P^h_{\mathbf{\Phi}|\mathbf{Y}}(\theta, \sigma|\mathbf{y})$.

For nonlinear forward maps **multimodality is very common**.

In our examples we use the t-walk Christen and Fox (2010), a self adjusted, generic MCMC for continuos distributions `http://www.cimat.mx/~jac/twalk/`. This is an affine invariant MCMC. Also, inspired by the t-walk, there is the emcee (the MCMC hammer) `http://dan.iel.fm/emcee/current/`.

Has no tuning parameters, and is suitable for mutimodal posteriors ... and is implemented in R, Python, C++, C and Matlab.

t-walk

Inclusive le hicieron su camiseta:

Bayesian model selection

Therefore, the comparison of models relies on the computation of the marginal likelihoods $P^h_{\mathbf{Y}}(\mathbf{y})$ which has been the object of a rich literature.

We use the Gelfand and Dey's estimator that recycles the evaluations of the MCMC, with basically no additional computational burden.

Main Result

We establish how to approximate the Bayes factors, without having the theoretical reference model, using solely the numerical solver approximation rates.

Main Result

Theorem

Assume that the numerical solver is such that the global error may be written as

$$E_h(t,\theta) = X_\theta^h(t) - X_\theta(t) = O(h^p),$$

where h is the stepsize of the method (ie. the solver is of order p). In addition, assume that the observation function f is differentiable on $\{X_\theta(t), \theta \in \Theta, t \in [0,T]\}$.

Then, there exists a constant $B(\mathbf{y}) \in \mathbb{R}$ (which does not depend on h) such that

$$\frac{P_{\mathbf{Y}}(\mathbf{y})}{P_{\mathbf{Y}}^h(\mathbf{y})} \simeq 1 + B(\mathbf{y})h^p.$$

Main Result

Corollary

Moreover If $\hat{g} = \int g(\phi) P_{\mathbf{\Phi}|\mathbf{Y}}(\phi|\mathbf{y}) d\phi$ *and* $\hat{g}^h = \int g(\phi) P^h_{\mathbf{\Phi}|\mathbf{Y}}(\phi|\mathbf{y}) d\phi$ *exists, then*

$$|\hat{g}^h - \hat{g}| = \frac{P_{\mathbf{Y}}(\mathbf{y})}{P^h_{\mathbf{Y}}(\mathbf{y})} B_g(\mathbf{y}) h^p = O(h^p),$$

for some constant $B_g(\mathbf{y})$ *(which does not depend on* h*).*

Numerical example: Logistic Growth

The dynamics are governed by the following differential equation

$$\frac{dX}{dt} = \lambda X(t)(K - X(t)), \quad X(0) = X_0 \tag{9}$$

with $r = \lambda K$ being the growth rate and K the carrying capacity e.g. $\lim_{t\to\infty} X(t) = K$.

Numerical example: Logistic Growth

Equation (9) has an explicit solution equal to

$$X(t) = \frac{KX_0 e^{\lambda K t}}{K + X_0(e^{\lambda K t} - 1)}.$$

We simulate two synthetic data sets with the error model $y_i = X(t_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, and the following parameters
$X(0) = 100, \quad \lambda = 1, \quad K = 1000$, $\sigma = 1$ or 30.

The datasets are plotted on Figure 1 for the two chosen values of σ. We consider 26 observations at times t_i regularly spaced between 0 and 10.

Numerical example: Logistic Growth

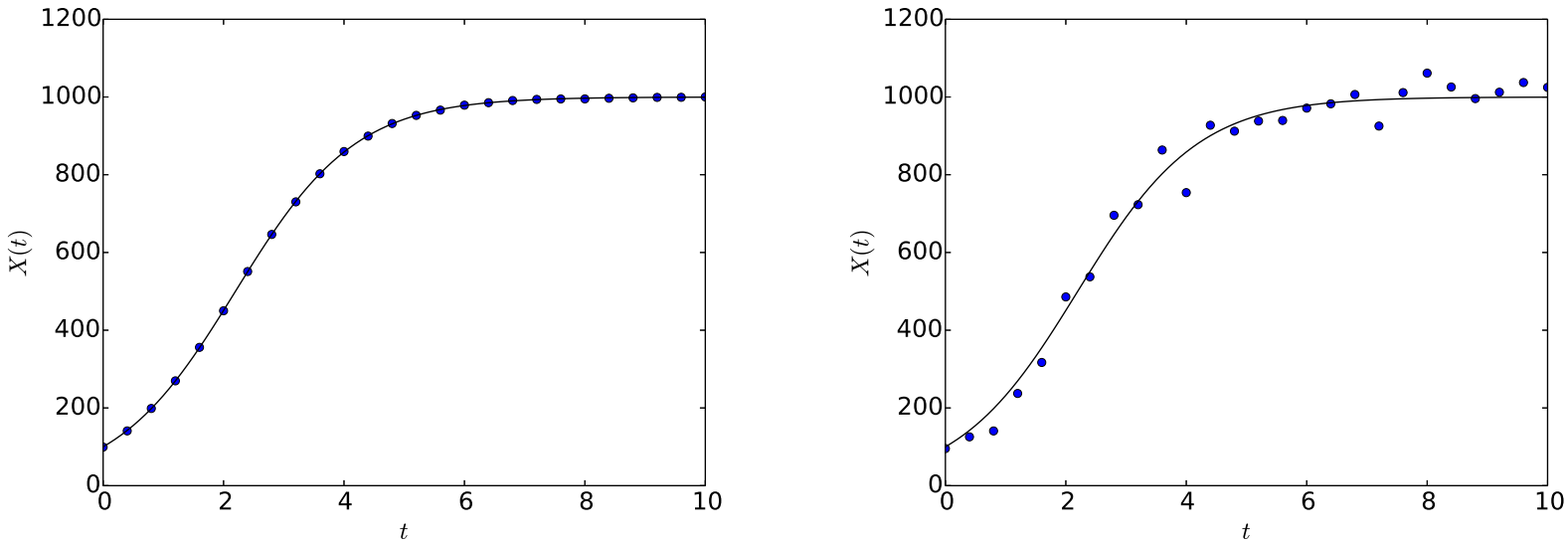


Figure: Synthetic data for the Logistic growth with $\lambda = 1$, $K = 1000$ and $\sigma = 1$ (left) or $\sigma = 30$ (right).

Numerical example: Logistic Growth

For this first toy example, K is taken as known and inference is concentrated on the single parameter λ; we consider a Gamma distribution for the prior on λ.

Numerical example: Logistic Growth, $\sigma = 1$

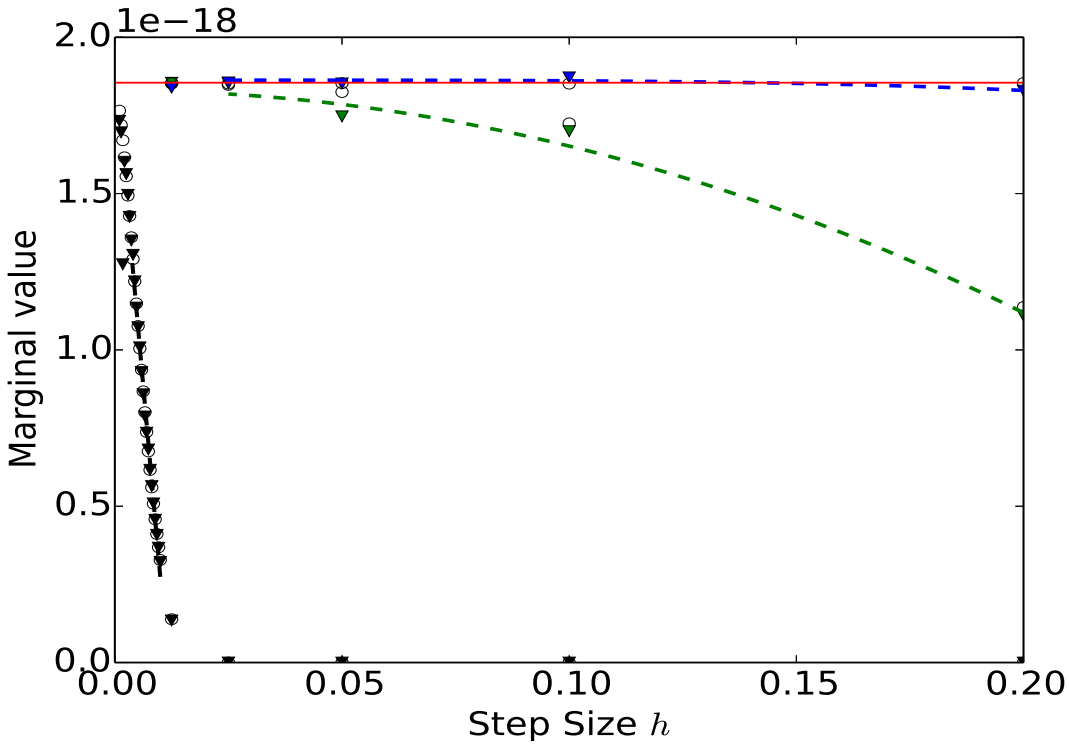


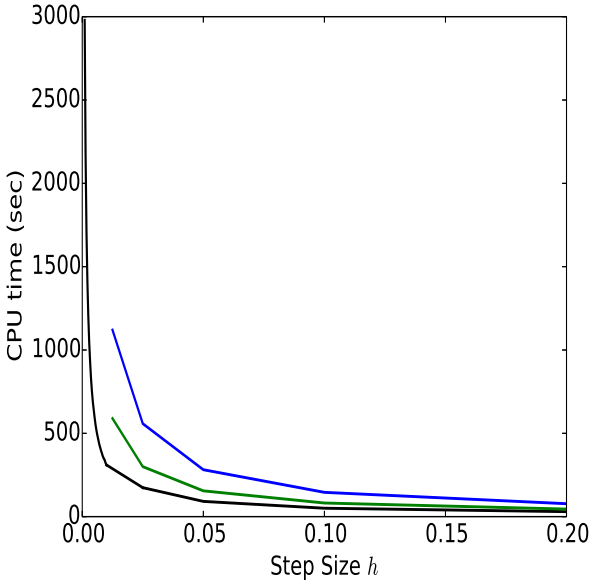
Figure: $P^h_{\mathbf{Y}}(\mathbf{y})$ for various step sizes, computed by numerical integration (solid thin lines) or estimated using the MCMC sample (triangles). In black, Runge-Kutta solver (RK) of order 1 (Euler), in green RK of order $p = 2$, in blue RK of order $p = 4$. Red line: true marginal $P_{\mathbf{Y}}(\mathbf{y})$ calculated using numerical integration on the analytic solution. Thick lines indicate the regression for estimated values for $\hat{P}^h_{\mathbf{Y}}(\mathbf{y}) = a + bh^p$ for the orders $p = 1, 2, 4$.

Numerical example: Logistic Growth, $\sigma = 1$

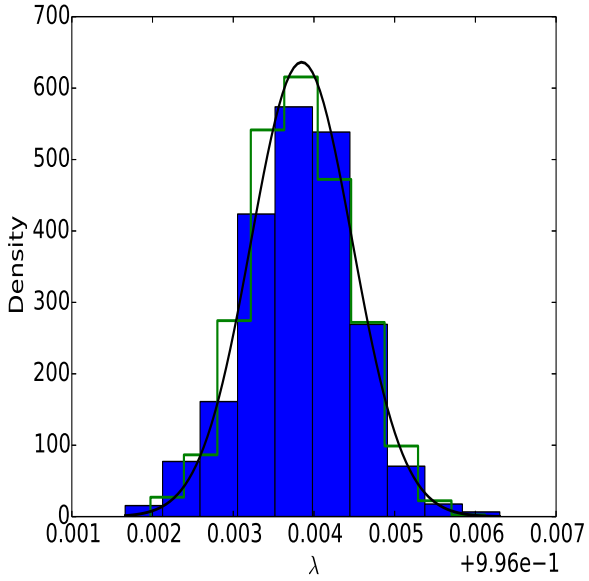
σ	$P_{\mathbf{Y}}(\mathbf{y})$	$\hat{P}_{\mathbf{Y}}(\mathbf{y})$
1	$1.854\,10^{-18}$	$1.862\,10^{-18}$
30	$1.638\,10^{-60}$	$1.699\,10^{-60}$

Table: Comparison of exact an estimated marginals for the Ringue-Kutta method of order 4.

Numerical example: Logistic Growth, $\sigma = 1$



(a)



(b)

Figure: (a) CPU time for various step values h_k and $p = 1, 2, 4$, relative to 10,000 iterations of the MCMC. (b) Posterior distribution of λ the for RK4 solver, $p = 4$, for step sizes $h = 0.01$ and $h = 0.05$ (histograms) and exact posterior (black density). 10,000 iterations of the MCMC took 17 min for $h = 0.01$ and 2 min for $h = 0.05$; a 90% reduction in CPU time with no noticeable difference in the resulting posterior distribution.

Numerical example, $\sigma = 30$

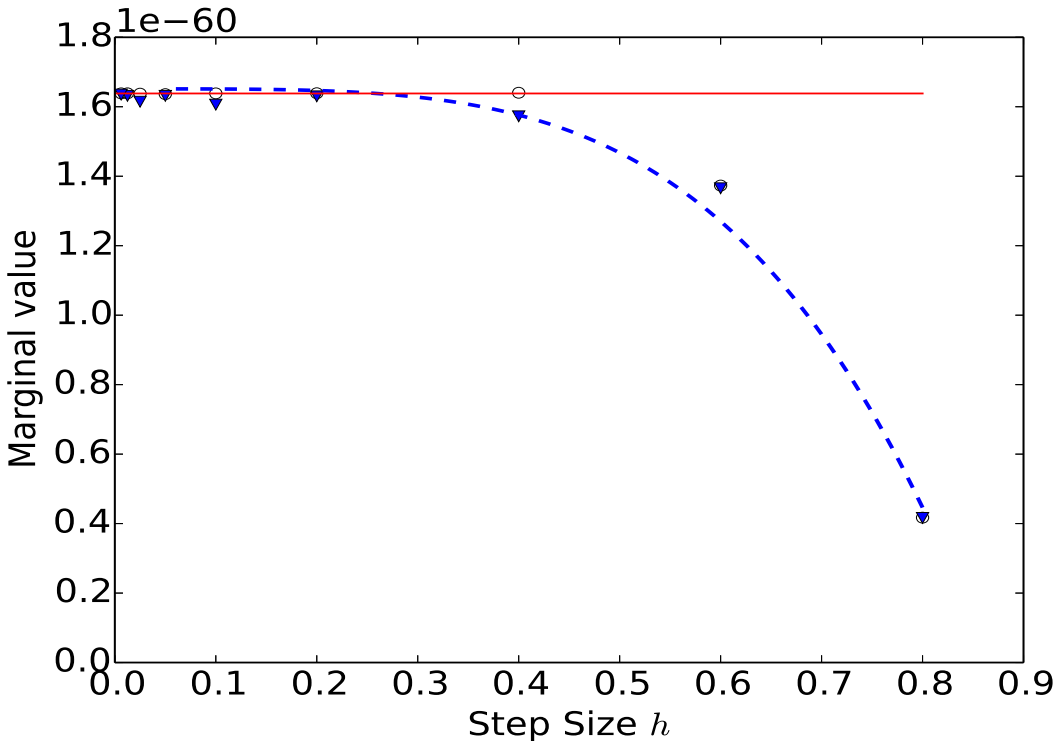
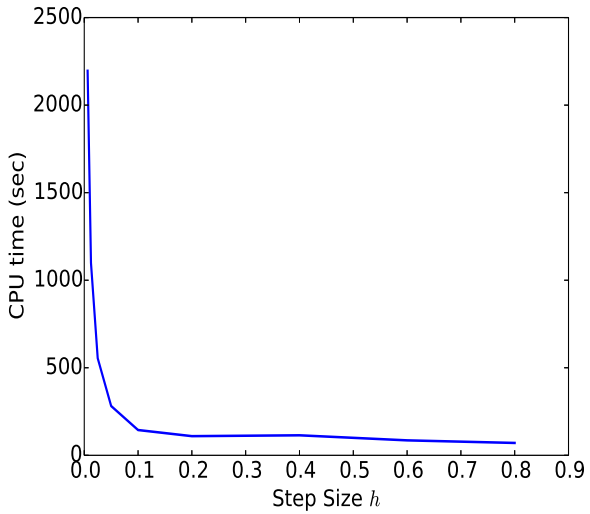
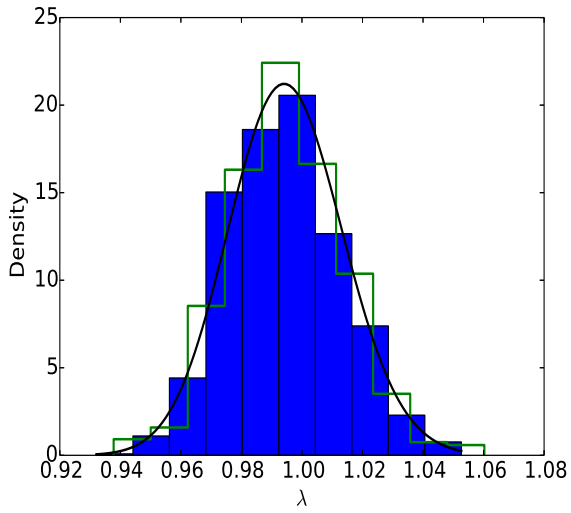


Figure: $P^h_{\mathbf{Y}}(\mathbf{y})$ for various step sizes, both exact (solid thin lines, using numerical integration) and estimated using the MCMC sample (triangles). We use a Rungue-Kutta solver of order 4 (classical RK4, blue), only. Red line: true marginal $P_{\mathbf{Y}}(\mathbf{y})$ calculated using numerical integration on the analytic solution. Thick lines indicate the regression for estimated values for $\hat{P}^h_{\mathbf{Y}}(\mathbf{y}) = a + bh^p$ for the order $p = 4$.

Numerical example, $\sigma = 30$



(a)



(b)

Figure: (a) Corresponding CPU time, relative to 10,000 iterations of the MCMC. (b) Posterior distribution of λ the for RK4 solver, $p = 4$, for step sizes $h = 0.00625$ and $h = 0.1$ (histograms; and exact posterior, black density). The former takes 36 min and the latter 2.5 min.

Numerical examples: A Diabetes minimal model

$$\frac{dG}{dt} = (L - I)\, G + \frac{D}{\theta_2}, \tag{10}$$

$$\frac{dI}{dt} = \theta_0 \left(\frac{G}{G_b} - 1 \right)^{+} - \frac{I}{a}, \tag{11}$$

$$\frac{dL}{dt} = \theta_1 \left(1 - \frac{G}{G_b} \right)^{+} - \frac{L}{b}, \tag{12}$$

$$\frac{dD}{dt} = -\frac{D}{\theta_2}. \tag{13}$$

Numerical examples: A Diabetes minimal model

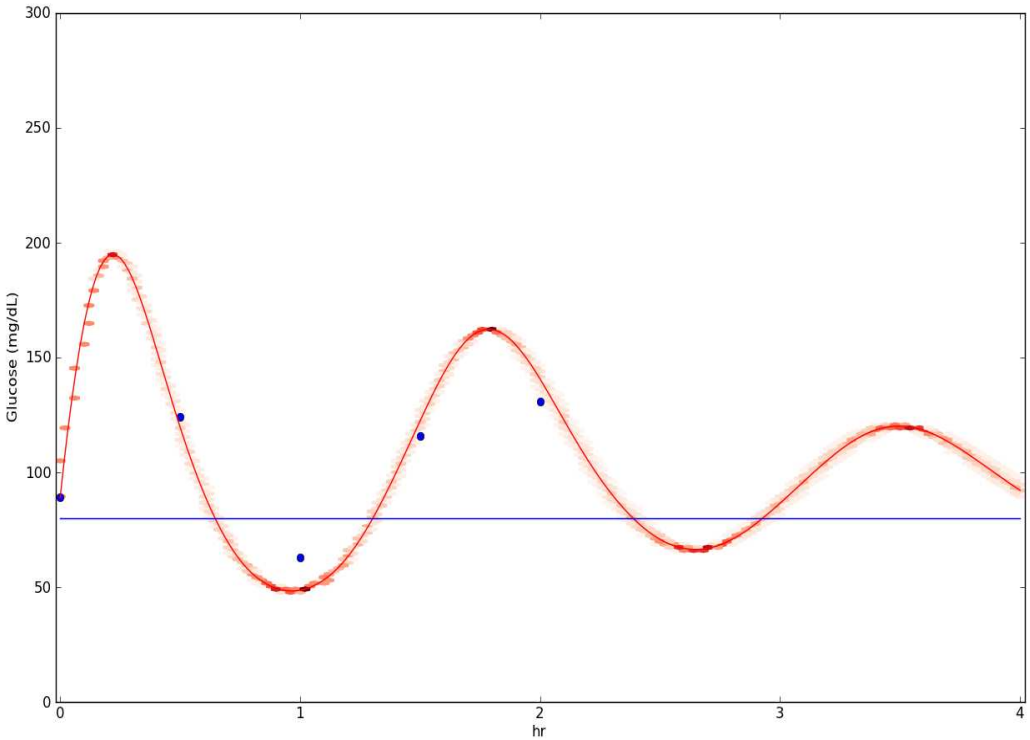


Figure: OGTT test performed in an obese male adult, with glucose measurents taken every 30 min up to 2 hr. Note the oscilating nature of the data, typically of a not well control Insulin-Glucose system. Both θ_0 and θ_1 have large values in comparison to normal subjects. The MAP model is shown in red, along with draws from the posterior predictive distribution shown in the shaded areas.

Numerical examples: A Diabetes minimal model

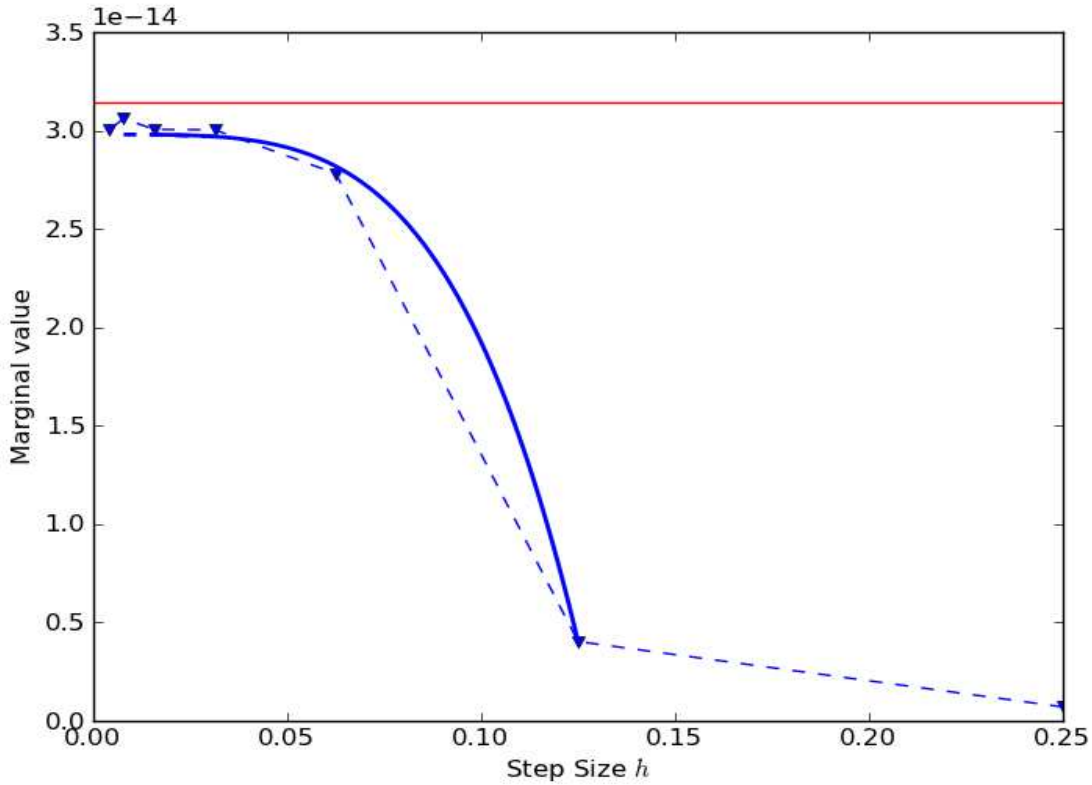
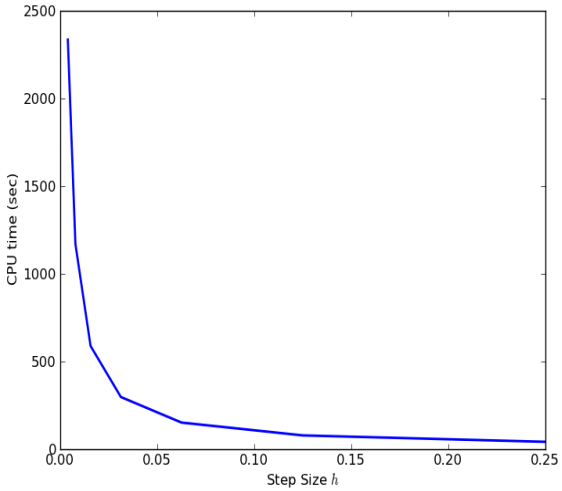
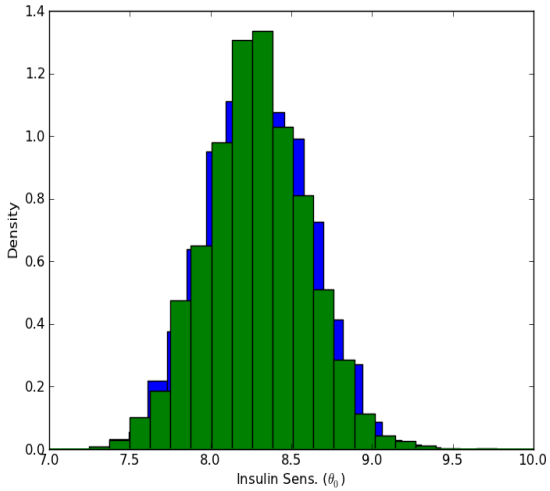


Figure: Bayes Factors study for the Diabetes minimal model. An order 4 Runge-Kutta solver was used to produce marginal values $P^h_{\mathbf{Y}}(\mathbf{Y})$ for step sizes as show. The red line in (a) is the numerical integration approximation of $P^h_{\mathbf{Y}}(\mathbf{Y})$ using step size $0.25 \cdot 2^{-7}$ (smallest step size used) while the triangles are Monte Carlo estimates; these seem to slightly underestimate the former. The solid blue line is a regression model $a + bh^4$ estimate using step sizes marginal estimates from $0.25 \cdot 2^{-1}$ to $0.25 \cdot 2^{-4}$ only.

Numerical examples: A Diabetes minimal model



(a)



(b)

Figure: (a) Corresponding CPU times for various step sizes. In (b) we compare the resulting posterior with step size $0.25 \cdot 2^{-3}$ and $0.25 \cdot 2^{-7}$ showing basically no difference and resulting in a near 90% reduction in CPU evaluation time.

Discussion

- We advance on some theoretical aspects of the Bayesian analysis of ODE systems.
- Basing our comparison on theoretical vs approximate posterior on the use of Bayes Factors, which is the natural tool to comparing models in a Bayesian context.
- We contribute to the intuitive idea that the ODE solver approximation error should be put in the perspective of the observational error. Our results establish a consistency in order accuracy for the solver and for the posterior distribution, considering BF's.
- 90% CPU was saved using **a less accurate solver**, that nevertheless **reaches the same, basically error less, results**.

Discussion

- We advance on some theoretical aspects of the Bayesian analysis of ODE systems.
- Basing our comparison on theoretical vs approximate posterior on the use of Bayes Factors, which is the natural tool to comparing models in a Bayesian context.
- We contribute to the intuitive idea that the ODE solver approximation error should be put in the perspective of the observational error. Our results establish a consistency in order accuracy for the solver and for the posterior distribution, considering BF's.
- 90% CPU was saved using **a less accurate solver**, that nevertheless **reaches the same, basically error less, results**.

Discussion

- We advance on some theoretical aspects of the Bayesian analysis of ODE systems.
- Basing our comparison on theoretical vs approximate posterior on the use of Bayes Factors, which is the natural tool to comparing models in a Bayesian context.
- We contribute to the intuitive idea that the ODE solver approximation error should be put in the perspective of the observational error. Our results establish a consistency in order accuracy for the solver and for the posterior distribution, considering BF's.
- 90% CPU was saved using **a less accurate solver**, that nevertheless **reaches the same, basically error less, results**.

Discussion

- We advance on some theoretical aspects of the Bayesian analysis of ODE systems.
- Basing our comparison on theoretical vs approximate posterior on the use of Bayes Factors, which is the natural tool to comparing models in a Bayesian context.
- We contribute to the intuitive idea that the ODE solver approximation error should be put in the perspective of the observational error. Our results establish a consistency in order accuracy for the solver and for the posterior distribution, considering BF's.
- 90% CPU was saved using **a less accurate solver**, that nevertheless **reaches the same, basically error less, results**.

Discussion

- Our results would also need to be stated for multiple dimension observation functions f.
- and tested in higher dimension parameter vector.
- Also, a generalization of the result to PDE's would be very useful.

Discussion

- Our results would also need to be stated for multiple dimension observation functions f.
- and tested in higher dimension parameter vector.
- Also, a generalization of the result to PDE's would be very useful.

Discussion

- Our results would also need to be stated for multiple dimension observation functions f.
- and tested in higher dimension parameter vector.
- Also, a generalization of the result to PDE's would be very useful.

¡Gracias!

¡GRACIAS!

Acknowledgments

We thank Dr Silvia Quintana for kindly providing the OGTT data used here. MAC and JAC would like to acknowledge financial support from Fondo Mixto de Fomento a la Investigación Científica y Tecnológica, CONACYT-Gobierno del Estado de Guajanuato, GTO-2011-C04-168776.

References

Christen, J. and C. Fox (2010). A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Analysis 5*(2), 263–282.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky (1999). Bayesian model averaging: A tutorial. *Statistical science 14*(4), 382–401.

Schwab, C. and A. M. Stuart (2012). Sparse deterministic approximation of bayesian inverse problems. *Inverse Problems 28*(4).

References

Christen, J. and C. Fox (2010). A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Analysis 5*(2), 263–282.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky (1999). Bayesian model averaging: A tutorial. *Statistical science 14*(4), 382–401.

Schwab, C. and A. M. Stuart (2012). Sparse deterministic approximation of bayesian inverse problems. *Inverse Problems 28*(4).

References

Christen, J. and C. Fox (2010). A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Analysis 5*(2), 263–282.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky (1999). Bayesian model averaging: A tutorial. *Statistical science 14*(4), 382–401.

Schwab, C. and A. M. Stuart (2012). Sparse deterministic approximation of bayesian inverse problems. *Inverse Problems 28*(4).