Probabilistic ABC: accelerating ABC using Gaussian processes

Richard Wilkinson

School of Mathematical Sciences
University of Nottingham

r.d.wilkinson@nottingham.ac.uk

Robotics Research Group, University of Oxford
August 2013

The need for simulation based methods

Rohrlich (1991): Computer simulation is

> *'a key milestone somewhat comparable to the milestone that started the empirical approach (Galileo) and the deterministic mathematical approach to dynamics (Newton and Laplace)'*

The need for simulation based methods

Rohrlich (1991): Computer simulation is

> *'a key milestone somewhat comparable to the milestone that started the empirical approach (Galileo) and the deterministic mathematical approach to dynamics (Newton and Laplace)'*

Challenges for statistics:
How do we make inferences about the world from a simulation of it?

The need for simulation based methods

Rohrlich (1991): Computer simulation is

> *'a key milestone somewhat comparable to the milestone that started the empirical approach (Galileo) and the deterministic mathematical approach to dynamics (Newton and Laplace)'*

Challenges for statistics:
How do we make inferences about the world from a simulation of it?

- how do we relate simulators to reality? (model error)
- how do we estimate tunable parameters? (calibration)
- how do we deal with computational constraints? (stat. comp.)
- how do we make uncertainty statements about the world that combine models, data and their corresponding errors? (UQ)

There is an inherent a lack of quantitative information on the uncertainty surrounding a simulation - unlike in physical experiments.

Calibration

Focus on simulator calibration:

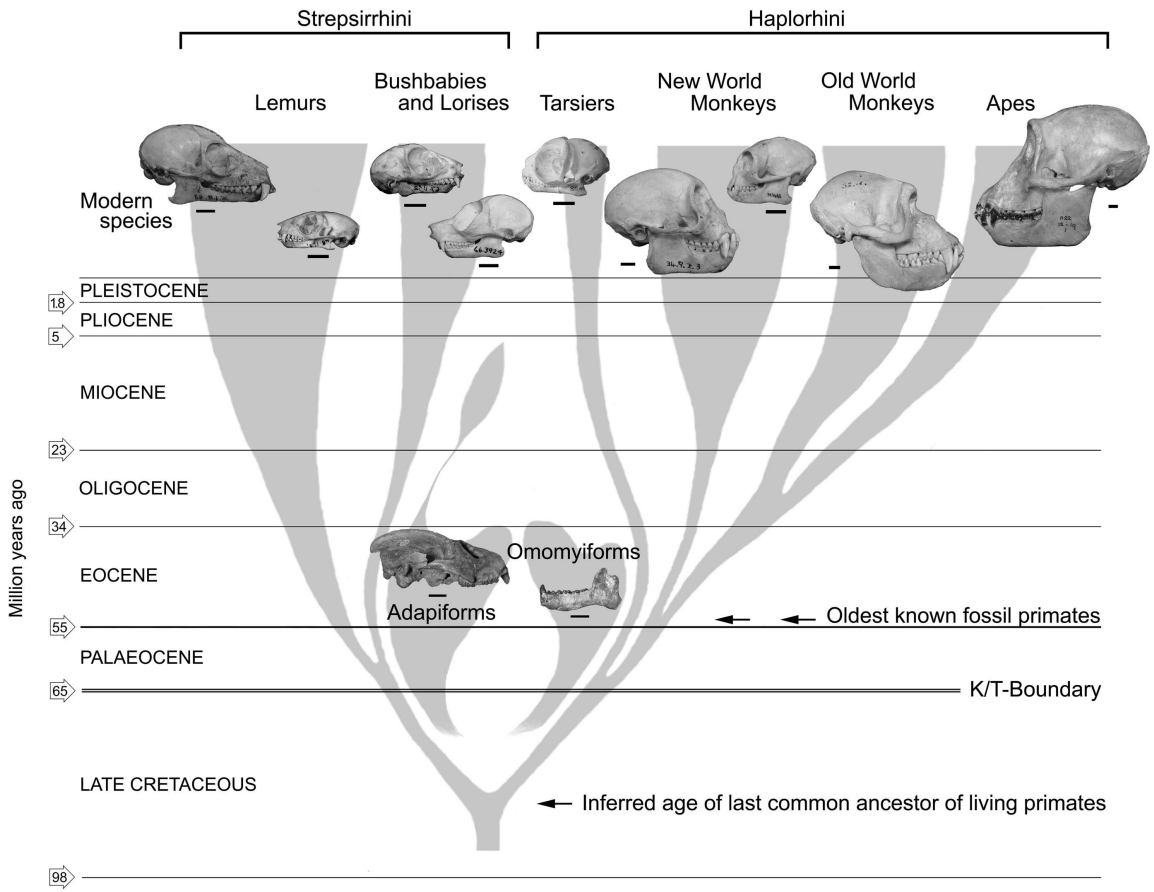
- For most simulators we specify parameters θ and i.c.s and the simulator, $f(\theta)$, generates output X.
- We are interested in the inverse-problem, i.e., observe data D, want to estimate parameter values θ that explain this data.

For Bayesians, this is a question of finding the posterior distribution

$$\pi(\theta|\mathcal{D}) \propto \pi(\theta)\pi(\mathcal{D}|\theta)$$

$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

The likelihood isn't just the simulator pdf



Basic idea and notation

- Suppose we want to find posterior distribution

$$\pi(\theta|D) = \frac{L(\theta)\pi(\theta)}{\pi(D)}$$

  where $L(\theta) = \pi(D|\theta)$ is the likelihood function.
- Suppose $L(\theta)$ is unknown, but we have estimates of its value at a small number of locations $\mathcal{C} = \{\theta_i, \hat{L}(\theta_i)\}_{i=1}^{N}$
- Build a Gaussian process model of $L(\theta)$ using $\mathcal{C}$.
- Find the posterior $\pi(\theta|\mathcal{D}, \mathcal{C})$

Intractable likelihoods

A Bayesian inference problem is intractable if the likelihood function

$$L(\theta) = \pi(D|\theta)$$

is unknown (even up to a normalising constant), I.e., if the distribution of the simulator, $f(\theta)$, run at θ is unknown.

- this is worse than the usual normalising constant intractability, or the double intractability of Murray and Ghahramani.

Intractable likelihoods

A Bayesian inference problem is intractable if the likelihood function

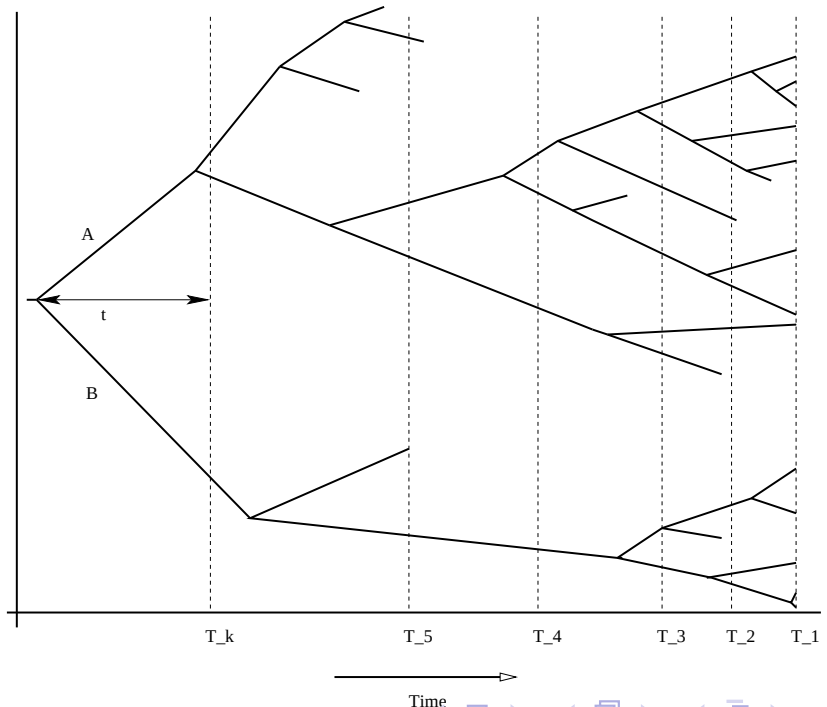
$$L(\theta) = \pi(D|\theta)$$

is unknown (even up to a normalising constant), I.e., if the distribution of the simulator, $f(\theta)$, run at θ is unknown.

- this is worse than the usual normalising constant intractability, or the double intractability of Murray and Ghahramani.

Example:

The density of the cumulative process of a branching process is unknown in general.



Approximate Bayesian Computation (ABC)

Approximate Bayesian Computation (ABC)

ABC algorithms are a collection of Monte Carlo methods used for calibrating simulators

- they do not require explicit knowledge of the likelihood function
- inference is done using simulation from the model (they are 'likelihood-free').

ABC methods have become popular in the biological sciences and versions of the algorithm exist in most modelling communities.

Approximate Bayesian Computation (ABC)

ABC algorithms are a collection of Monte Carlo methods used for calibrating simulators

- they do not require explicit knowledge of the likelihood function
- inference is done using simulation from the model (they are 'likelihood-free').

ABC methods have become popular in the biological sciences and versions of the algorithm exist in most modelling communities.

ABC methods can be crude but they have an important role to play.

Likelihood-Free Inference

Rejection Algorithm

- Draw θ from prior $\pi(\cdot)$
- Accept θ with probability $\pi(\mathcal{D} \mid \theta)$

Accepted θ are independent draws from the posterior distribution, $\pi(\theta \mid \mathcal{D})$.

Likelihood-Free Inference

Rejection Algorithm

- Draw θ from prior $\pi(\cdot)$
- Accept θ with probability $\pi(\mathcal{D} \mid \theta)$

Accepted θ are independent draws from the posterior distribution, $\pi(\theta \mid \mathcal{D})$.
If the likelihood, $\pi(\mathcal{D}|\theta)$, is unknown:

'Mechanical' Rejection Algorithm

- Draw θ from $\pi(\cdot)$
- Simulate $X \sim f(\theta)$ from the computer model
- Accept θ if $\mathcal{D} = X$, i.e., if computer output equals observation

The acceptance rate is $\mathbb{P}(\mathcal{D})$: the number of runs to get n observations is negative binomial, with mean $\frac{n}{\mathbb{P}(\mathcal{D})}$: $\Rightarrow$ Bayes Factors!

Uniform ABC algorithms

Uniform ABC

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(\mathcal{D}, X) \leq \epsilon$

For reasons that will become clear later, call this *Uniform ABC*.

Uniform ABC algorithms

Uniform ABC

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(\mathcal{D}, X) \leq \epsilon$

For reasons that will become clear later, call this *Uniform ABC*.

- As $\epsilon \to \infty$, we get observations from the prior, $\pi(\theta)$.
- If $\epsilon = 0$, we generate observations from $\pi(\theta \mid \mathcal{D})$

ϵ reflects the tension between computability and accuracy.

The hope is that $\pi_{ABC}(\theta) \approx \pi(\theta|D, PSH)$ for ϵ small, where PSH='perfect simulator hypothesis'

Uniform ABC algorithms

Uniform ABC

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(\mathcal{D}, X) \leq \epsilon$

For reasons that will become clear later, call this *Uniform ABC*.

- As $\epsilon \to \infty$, we get observations from the prior, $\pi(\theta)$.
- If $\epsilon = 0$, we generate observations from $\pi(\theta \mid \mathcal{D})$

ϵ reflects the tension between computability and accuracy.

The hope is that $\pi_{ABC}(\theta) \approx \pi(\theta|D, PSH)$ for ϵ small, where PSH='perfect simulator hypothesis'
There are uniform ABC-MCMC, ABC-SMC, ABC-EM, ABC-EP, ABC-MLE algorithms, etc.

ABC choices

Most of the early ABC developments have been in an algorithmic tradition.

1. Find a good metric, ρ - e.g., L_2 norm
2. Find a good ϵ - e.g., best 1% of simulations?
3. Find a good summary $S(D)$

The choices made are usually not motivated by modelling considerations.

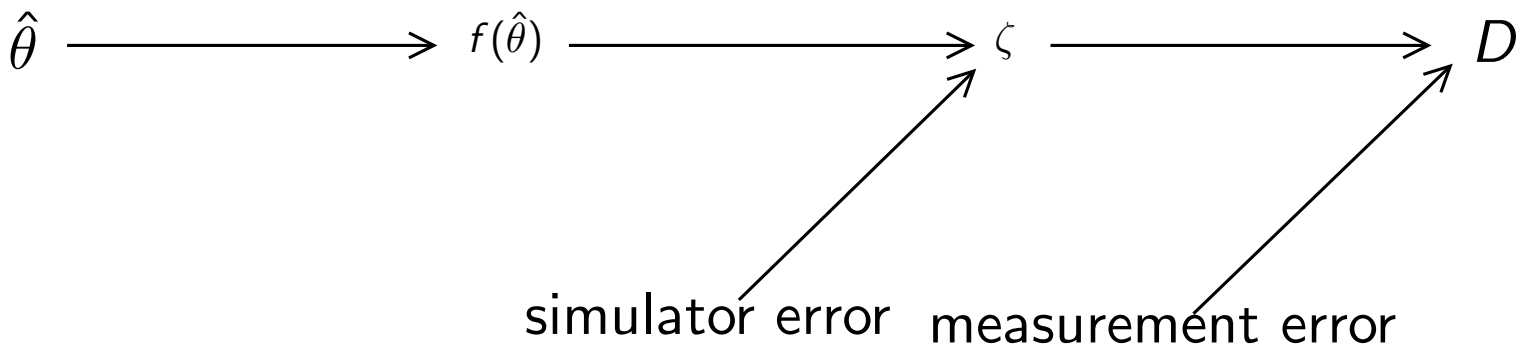
Poor choices for any of these aspects can have unintended consequences.

Calibration framework

There is a probabilistic interpretation of ABC.

Consider the Bayesian calibration framework from the computer experiment literature:

- Relate the best-simulator run ($X = f(\hat{\theta})$) to reality ζ
- Relate reality ζ to the observations D.



See, for example, Kennedy and O'Hagan (2001) or Goldstein and Rougier (2009).

Calibration framework

Mathematically, we can write the likelihood as

$$L(\theta) = \pi(D|\theta) = \int \pi(D|x)\pi(x|\theta)\mathrm{dx}$$

where

- $\pi(D|x)$ is a pdf relating the simulator output to reality - call it the *acceptance kernel*.
- $\pi(x|\theta)$ is the likelihood function of the simulator (ie not relating to reality)

The posterior is

$$\pi(\theta|D) = \frac{1}{Z} \int \pi(D|x)\pi(x|\theta)\mathrm{dx}.\ \pi(\theta)$$

where $Z = \int\int \pi(D|x)\pi(x|\theta)\mathrm{dx}\pi(\theta)\mathrm{d}\theta$

How does ABC relate to calibration?

Wilkinson 2008/2013

To simplify matters, we can work in joint (θ, x) space

$$\pi(\theta, x|D) = \frac{\pi(D|x)\pi(x|\theta)\pi(\theta)}{Z}$$

Sample from this using the rejection algorithm with instrumental distribution

$$g(\theta, x) = \pi(x|\theta)\pi(\theta)$$

Generalized ABC (GABC)

Wilkinson 2008, Fearnhead and Prangle 2012

The rejection algorithm then becomes

Generalized rejection ABC (Rej-GABC)

1. $\theta \sim \pi(\theta)$ and $X \sim \pi(x|\theta)$ (ie $(\theta, X) \sim g(\cdot)$)
2. Accept (θ, X) if

$$U \sim U[0,1] \leq \frac{\pi_{ABC}(\theta, x)}{Mg(\theta, x)} = \frac{\pi(D|X)}{\max_x \pi(D|x)}$$

Generalized ABC (GABC)

Wilkinson 2008, Fearnhead and Prangle 2012

The rejection algorithm then becomes

Generalized rejection ABC (Rej-GABC)

1. $\theta \sim \pi(\theta)$ and $X \sim \pi(x|\theta)$ (ie $(\theta, X) \sim g(\cdot)$)
2. Accept (θ, X) if

$$U \sim U[0,1] \leq \frac{\pi_{ABC}(\theta, x)}{Mg(\theta, x)} = \frac{\pi(D|X)}{\max_x \pi(D|x)}$$

In uniform ABC we take

$$\pi(D|X) = \begin{cases} 1 & \text{if } \rho(D, X) \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

this reduces the algorithm to

2' Accept θ ifF $\rho(D, X) \leq \epsilon$

ie, we recover the *uniform* ABC algorithm.

Uniform ABC algorithm

This allows us to interpret uniform ABC. Suppose $X, D \in \mathcal{R}$

Proposition

Accepted θ from the uniform ABC algorithm (with $\rho(D, X) = |D - X|$) are samples from the posterior distribution of θ given D where we assume $D = f(\theta) + e$ and that

$$e \sim U[-\epsilon, \epsilon]$$

In general, uniform ABC assumes that

$$D|x \sim U\{d : \rho(d, x) \leq \epsilon\}$$

We can think of this as assuming a uniform error term when we relate the simulator to the observations.

Uniform ABC algorithm

This allows us to interpret uniform ABC. Suppose $X, D \in \mathcal{R}$

Proposition

Accepted θ from the uniform ABC algorithm (with $\rho(D, X) = |D - X|$) are samples from the posterior distribution of θ given D where we assume $D = f(\theta) + e$ and that

$$e \sim U[-\epsilon, \epsilon]$$

In general, uniform ABC assumes that

$$D|x \sim U\{d : \rho(d, x) \leq \epsilon\}$$

We can think of this as assuming a uniform error term when we relate the simulator to the observations.

ABC gives 'exact' inference under a different model!

Wood (2010)

Key idea: introduce a synthetic Gaussian likelihood function for the simulator, and then use MCMC to find the posterior.

Wood 2010

Suppose our MCMC chain is currently at θ_i.

- Propose a move to θ' from some kernel
- Run the simulator n times at θ', giving realisations $X_1, \ldots, X_n$
- Summarize these to get summaries $S_1, \ldots, S_n$.
- Assume $S \sim N(\mu_{\theta'}, \Sigma_{\theta'})$, and estimate $\mu_{\theta'}$ and $\Sigma_{\theta'}$.
- Assign θ' likelihood $\phi(s^{obs}; \mu_{\theta'}, \Sigma_{\theta'})$ and accept or reject θ' according the MH acceptance ratio.

Wood (2010)

Key idea: introduce a synthetic Gaussian likelihood function for the simulator, and then use MCMC to find the posterior.

Wood 2010

Suppose our MCMC chain is currently at θ_i.

- Propose a move to θ' from some kernel
- Run the simulator n times at θ', giving realisations $X_1, \ldots, X_n$
- Summarize these to get summaries $S_1, \ldots, S_n$.
- Assume $S \sim N(\mu_{\theta'}, \Sigma_{\theta'})$, and estimate $\mu_{\theta'}$ and $\Sigma_{\theta'}$.
- Assign θ' likelihood $\phi(s^{obs}; \mu_{\theta'}, \Sigma_{\theta'})$ and accept or reject θ' according the MH acceptance ratio.

This is a GABC algorithm, using μ_θ and Σ_θ as the summary of $f(\theta)$, and assuming

$$\pi(D|S) = \exp(-\frac{1}{2}(D - \mu_\theta)^T \Sigma_\theta^{-1}(D - \mu_\theta))$$

It can be seen as accounting for the variability of the model run repeatedly at the same input, and then assuming the distribution is Gaussian.

Problems with Monte Carlo methods

Monte Carlo methods are generally guaranteed to succeed if we run them for long enough.

This guarantee comes at a cost.

- Most methods sample naively - they don't learn from previous simulations.
- They don't exploit known properties of the likelihood function, such as continuity
- They sample randomly, rather than using space filling designs.

This naivety can make a full analysis infeasible without access to a large amount of computational resource.

Problems with Monte Carlo methods

Monte Carlo methods are generally guaranteed to succeed if we run them for long enough.

This guarantee comes at a cost.

- Most methods sample naively - they don't learn from previous simulations.
- They don't exploit known properties of the likelihood function, such as continuity
- They sample randomly, rather than using space filling designs.

This naivety can make a full analysis infeasible without access to a large amount of computational resource.

If we are prepared to lose the guarantee of eventual success, we can exploit the continuity of the likelihood function to learn about its shape, and to dramatically improve the efficiency of our computations.

Likelihood estimation

The GABC framework assumes

$$\pi(D|\theta) = \int \pi(D|X)\pi(X|\theta)\mathrm{dX}$$
$$\approx \frac{1}{N}\sum \pi(D|X_i)$$

where $X_i \sim \pi(X|\theta)$. Or in Wood (2010),

$$\pi(D|\theta) = \phi(D; \mu_\theta, \Sigma_\theta)$$

Likelihood estimation

The GABC framework assumes

$$\pi(D|\theta) = \int \pi(D|X)\pi(X|\theta)\mathrm{dX}$$
$$\approx \frac{1}{N}\sum \pi(D|X_i)$$

where $X_i \sim \pi(X|\theta)$. Or in Wood (2010),

$$\pi(D|\theta) = \phi(D; \mu_\theta, \Sigma_\theta)$$

For many problems, we believe the likelihood is continuous and smooth, so that $\pi(D|\theta)$ is similar to $\pi(D|\theta')$ when $\theta - \theta'$ is small

We can model $L(\theta) = \pi(D|\theta)$ and use the model to find the posterior in place of running the simulator.

Example: Ricker Model

The Ricker model is one of the prototypic ecological models.

- used to model the fluctuation of the observed number of animals in some population over time
- It has complex dynamics and likelihood, despite its simple mathematical form.

Ricker Model

- Let N_t denote the number of animals at time t.

$$N_{t+1} = rN_t e^{-N_t + e_r}$$

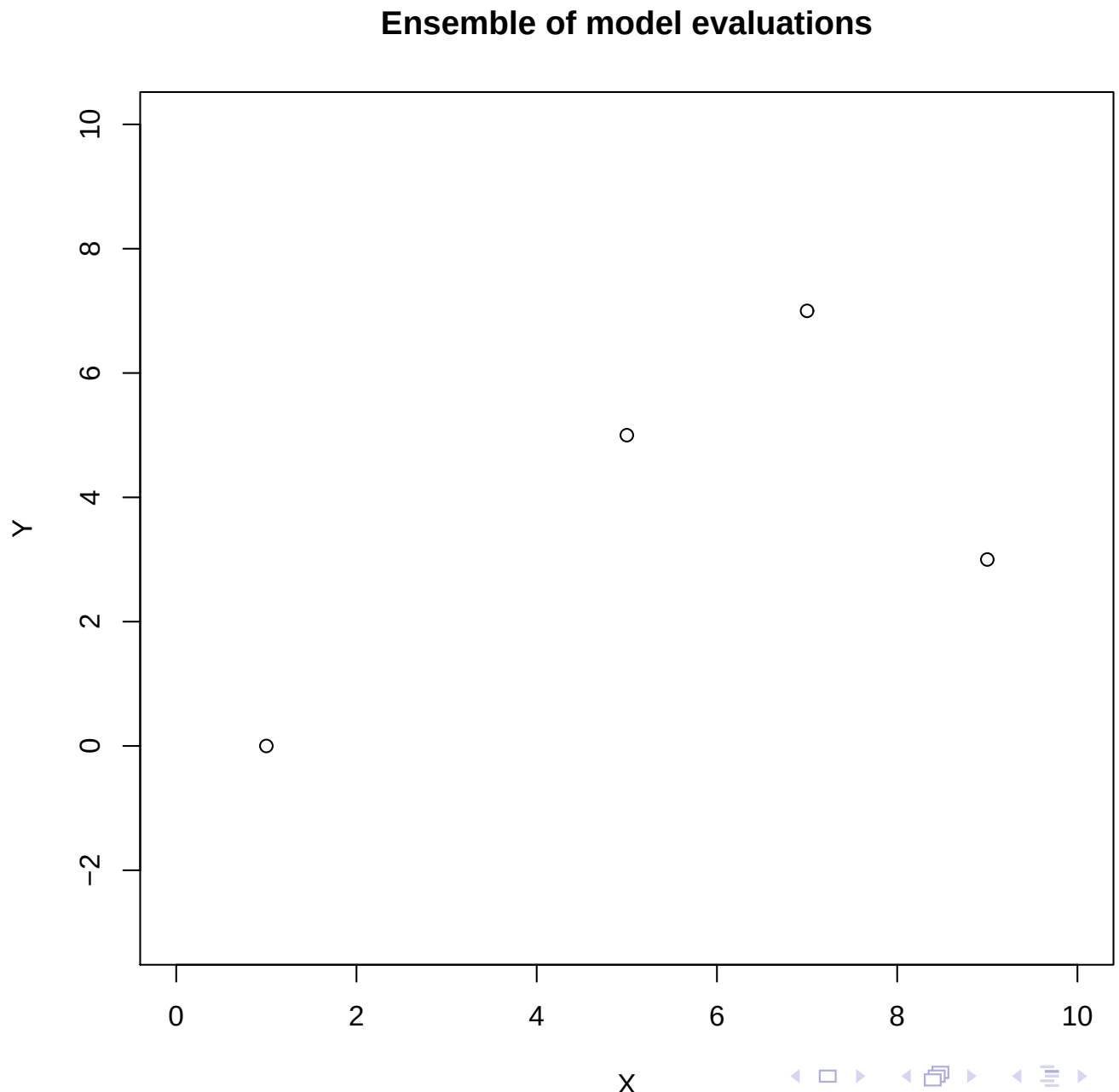
  where e_t are independent $N(0, \sigma_e^2)$ process noise

- Assume we observe counts y_t where

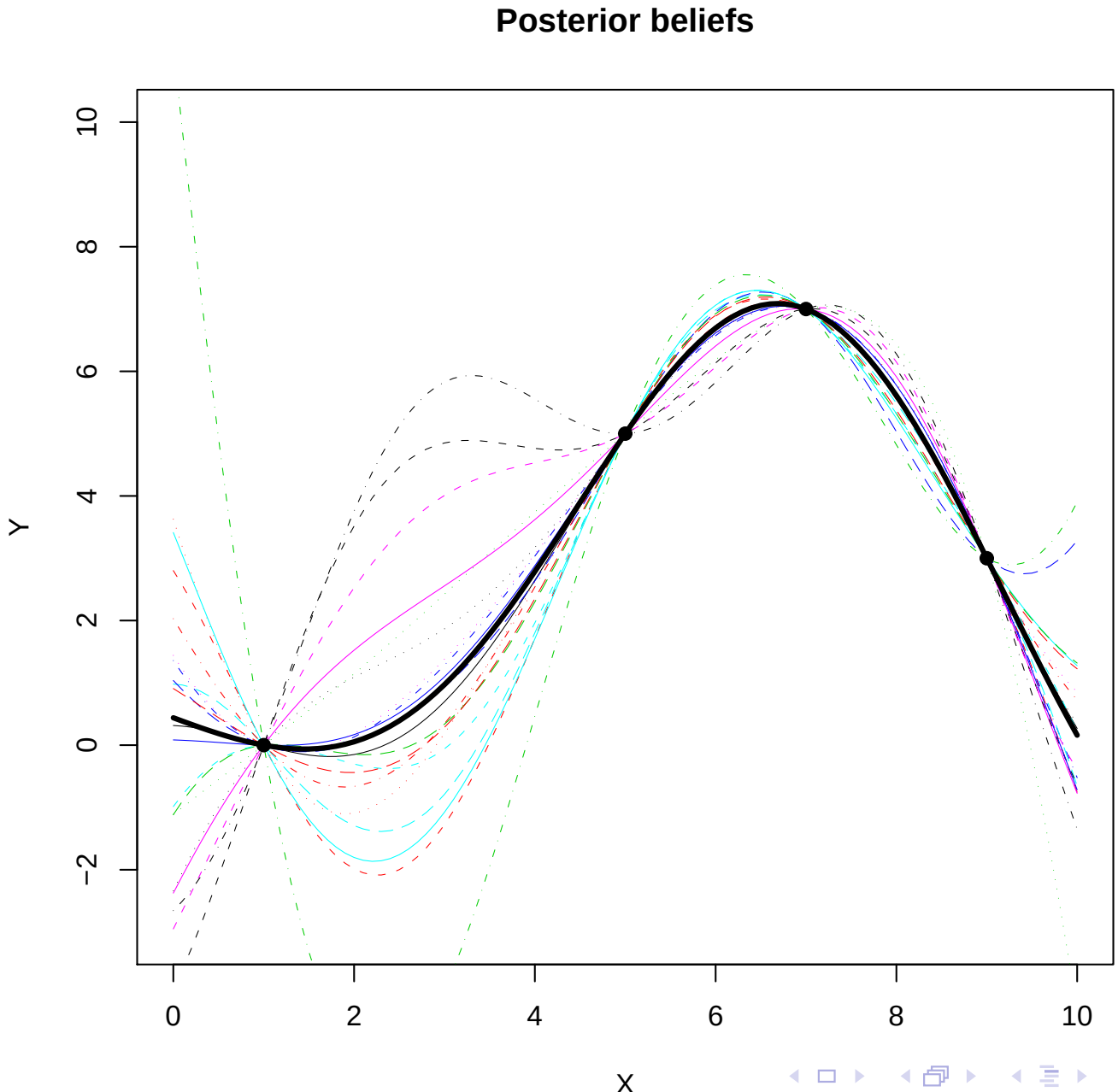
$$y_t \sim Po(\phi N_t)$$

Used in Wood to demonstrate the synthetic likelihood approach.

Gaussian Process Illustration



Gaussian Process Illustration



GP emulator choices

Let $L(\theta) = \pi(D|\theta)$. We model a transformation of the likelihood

$$G(\theta) = \Phi^{-1}(L(\theta))$$

typically $\Phi^{-1}(L) = \log(L)$ or $\log(-\log(L))$.

GP emulator choices

Let $L(\theta) = \pi(D|\theta)$. We model a transformation of the likelihood

$$G(\theta) = \Phi^{-1}(L(\theta))$$

typically $\Phi^{-1}(L) = \log(L)$ or $\log(-\log(L))$.

Use a Gaussian process (GP) prior

$$G(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$$

with $m(\theta) = h(\theta)^T \beta$.

$$G(\theta) = h(\theta)\beta + u(\theta)$$

$$\text{emulator} = \text{mean structure} + \text{residual}$$

GP emulator choices

Let $L(\theta) = \pi(D|\theta)$. We model a transformation of the likelihood

$$G(\theta) = \Phi^{-1}(L(\theta))$$

typically $\Phi^{-1}(L) = \log(L)$ or $\log(-\log(L))$.

Use a Gaussian process (GP) prior

$$G(\cdot) \sim GP(m(\cdot), k(\cdot,\cdot))$$

with $m(\theta) = h(\theta)^T\beta$.

$$G(\theta) = h(\theta)\beta + u(\theta)$$

$$\text{emulator} = \text{mean structure} + \text{residual}$$

We let the mean function $h(\theta)$ include up to quadratic polynomial terms, as typically we know

$$\log(L(\theta)) \to -\infty \text{ as } \theta \to \pm\infty$$

GP emulator choices II

Typically, use a squared-exponential covariance function

$$k(\theta, \theta') = \sigma^2 \exp\left(-(\theta - \theta')\Sigma^{-1}(\theta - \theta')\right)$$

where Σ is a matrix of length-scales (typically $\Sigma = \text{diag}(\lambda)$).

GP emulator choices II

Typically, use a squared-exponential covariance function

$$k(\theta, \theta') = \sigma^2 \exp\left(-(\theta - \theta')\Sigma^{-1}(\theta - \theta')\right)$$

where Σ is a matrix of length-scales (typically $\Sigma = \mathrm{diag}(\lambda)$).

Assume we have observations of the likelihood: Let $\Theta = \{\theta_i\}_{i=1}^n$ be a design on the parameter space, and let $\mathbf{G} = (G_1, \ldots, G_n)^T$ be the corresponding estimated log-likelihood values. Then our training set is

$$\mathcal{C} = \{\mathbf{\Theta}, \mathbf{G}\}$$

Assume observations $G_i \sim \Phi^{-1}(L(\theta_i)) + N(0, \tau)$, i.e., unbiased estimates with Gaussian error.

GP emulator choices II

Typically, use a squared-exponential covariance function

$$k(\theta, \theta') = \sigma^2 \exp\left(-(\theta - \theta')\Sigma^{-1}(\theta - \theta')\right)$$

where Σ is a matrix of length-scales (typically $\Sigma = \text{diag}(\lambda)$).

Assume we have observations of the likelihood: Let $\Theta = \{\theta_i\}_{i=1}^n$ be a design on the parameter space, and let $\mathbf{G} = (G_1, \ldots, G_n)^T$ be the corresponding estimated log-likelihood values. Then our training set is

$$\mathcal{C} = \{\mathbf{\Theta}, \mathbf{G}\}$$

Assume observations $G_i \sim \Phi^{-1}(L(\theta_i)) + N(0, \tau)$, i.e., unbiased estimates with Gaussian error.

Then

$$\mathbf{G} \sim N(\mathbf{m}(\theta), K_y)$$

where

$$\{K_y\}_{i,j} = k(\theta_i, \theta_j) + \tau\delta_{i=j}$$

i.e., include a nugget term to represent the uncertainty in the observations.

GP updates

Given training set $\mathcal{C}$, the GP posterior is

$$G(\cdot)|\mathcal{C}, \Psi, \beta \sim GP(m^*(\cdot), k^*(\cdot, \cdot))$$

and if we give β a flat improper prior $\pi(\beta) \propto 1$, we can integrate out the dependence on β to find

$$G(\cdot)|\mathcal{C}, \Psi \sim GP(m^{**}(\cdot), k^{**}(\cdot, \cdot))$$

where $\Psi = \{\Sigma, \sigma^2\}$ are GP hyper parameters.

GP updates

Given training set $\mathcal{C}$, the GP posterior is

$$G(\cdot)|\mathcal{C}, \Psi, \beta \sim GP(m^*(\cdot), k^*(\cdot, \cdot))$$

and if we give β a flat improper prior $\pi(\beta) \propto 1$, we can integrate out the dependence on β to find

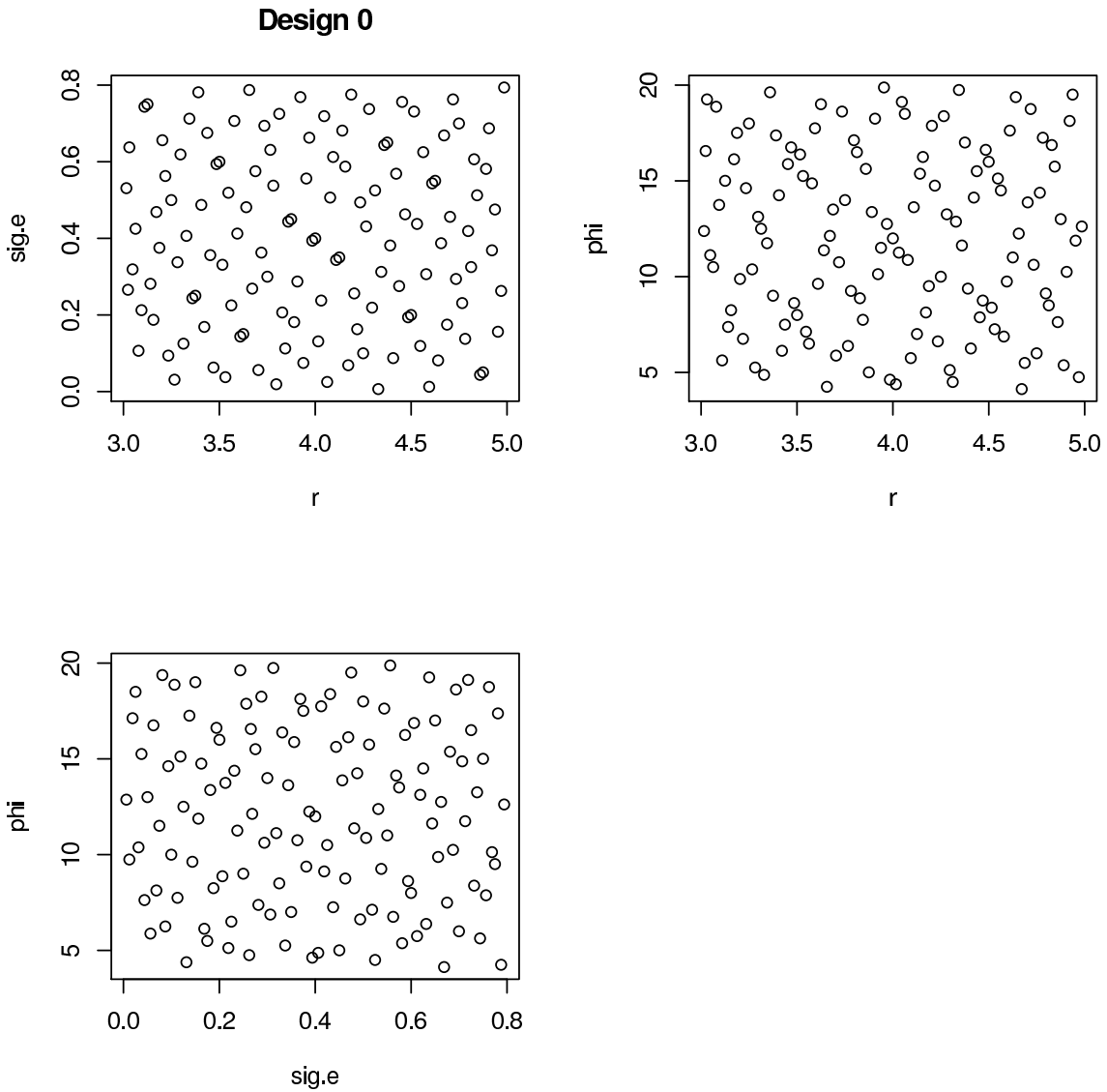
$$G(\cdot)|\mathcal{C}, \Psi \sim GP(m^{**}(\cdot), k^{**}(\cdot, \cdot))$$

where $\Psi = \{\Sigma, \sigma^2\}$ are GP hyper parameters.

There are no conjugate priors available for Σ or σ^2.

Design 1 - 128 pts

We use a Sobol sequence on the prior input space to find a design $\{\theta_i\}_{i=1}^{d}$. We estimate the likelihood at each point in the design, and aim to fit a GP model to estimate the likelihood at θ values not in the design.



History matching waves

The likelihood is too difficult to model, so we model the log-likelihood instead.

$$G(\theta) = \log L(\theta), \qquad \hat{L}(\theta_i) = \frac{1}{N} \sum \pi(D|X_i),\ X_i \sim \pi(X|\theta_i)$$

History matching waves

The likelihood is too difficult to model, so we model the log-likelihood instead.

$$G(\theta) = \log L(\theta), \qquad \hat{L}(\theta_i) = \frac{1}{N} \sum \pi(D|X_i),\ X_i \sim \pi(X|\theta_i)$$

However, the log-likelihood for a typical problem ranges across too wide a range of values, e.g., -10 near the mode, but essentially $-\infty$ at the extremes of the prior range.
Consequently, any Gaussian process model will struggle to model the log-likelihood across the entire input range.

History matching waves

The likelihood is too difficult to model, so we model the log-likelihood instead.

$$G(\theta) = \log L(\theta), \qquad \hat{L}(\theta_i) = \frac{1}{N} \sum \pi(D|X_i),\ X_i \sim \pi(X|\theta_i)$$

However, the log-likelihood for a typical problem ranges across too wide a range of values, e.g., -10 near the mode, but essentially $-\infty$ at the extremes of the prior range.
Consequently, any Gaussian process model will struggle to model the log-likelihood across the entire input range.

- To fix this we introduce the idea of waves, similar to those used in Michael Goldstein's approach to history-matching.
- In each wave, we build a GP model that can rule out large swathes of space as *implausible*.

Implausibility

We decide that θ is implausible if

$$m(\theta) + 3\sigma < \max_{\theta_i} \log L(\theta_i) - T$$

where $m(\theta)$ is the Gaussian process estimate of $\log \pi(D|\theta)$, and σ is the variance of the GP estimate.

Implausibility

We decide that θ is implausible if

$$m(\theta) + 3\sigma < \max_{\theta_i} \log L(\theta_i) - T$$

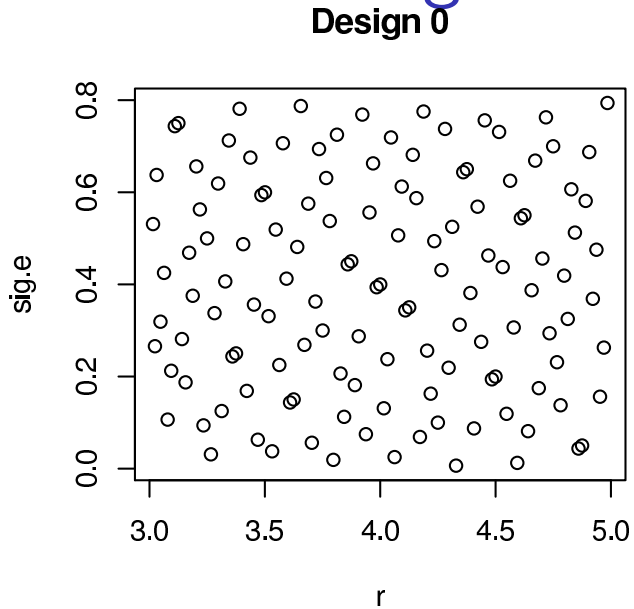
where $m(\theta)$ is the Gaussian process estimate of $\log \pi(D|\theta)$, and σ is the variance of the GP estimate.

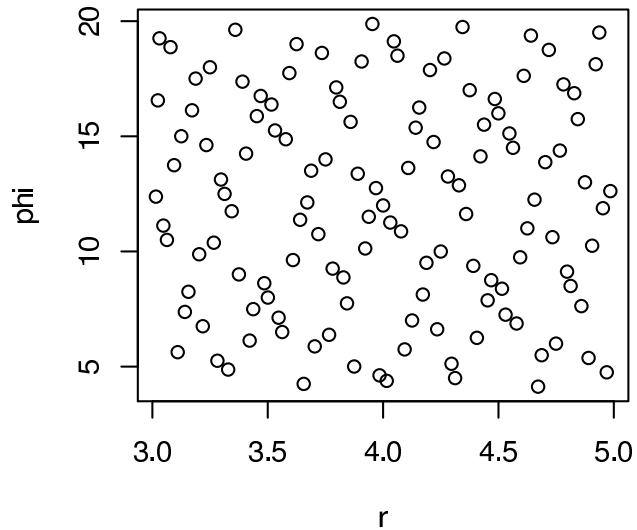
- We subtract a threshold value $T = 10$ for the Ricker model: a difference of 10 on the log scale between two likelihoods, means that assigning the θ with the smaller log-likelihood a posterior density of 0 (by saying it is implausible) is a good approximation.

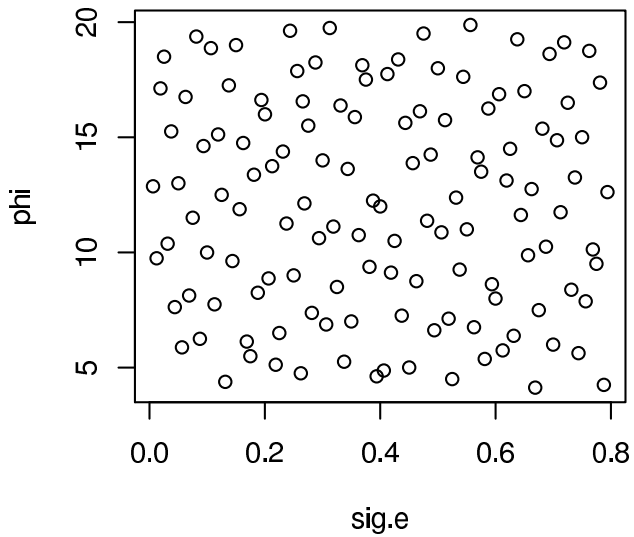
Difficulties

- This still wasn't enough in some problems, so for the first wave we model $\log(-\log L(\theta))$
- For the next wave, we begin by using the Gaussian processes from the previous waves to decide which parts of the input space are implausible.
- We then extend the design into the not-implaussible range and build a new Gaussian process
- This new GP will lead to a new definition of implausibility
- ...

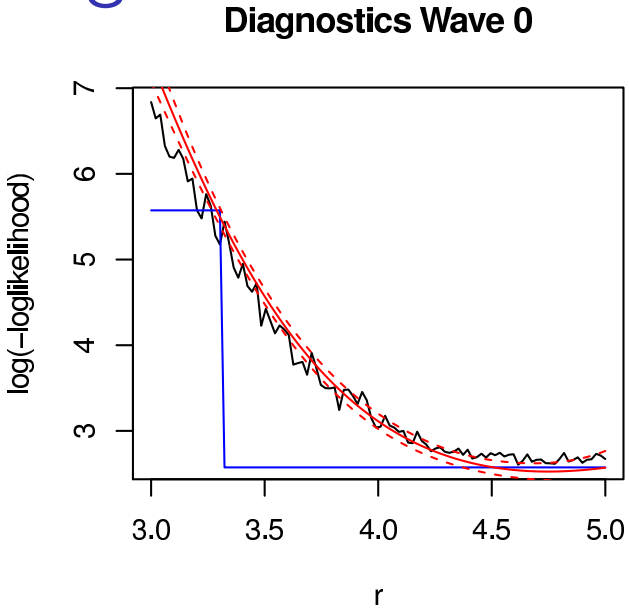
Results - Design 1 - 128 pts

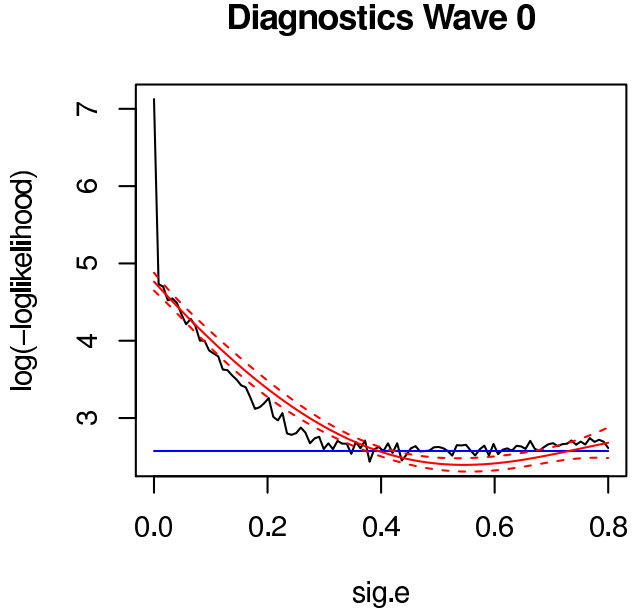


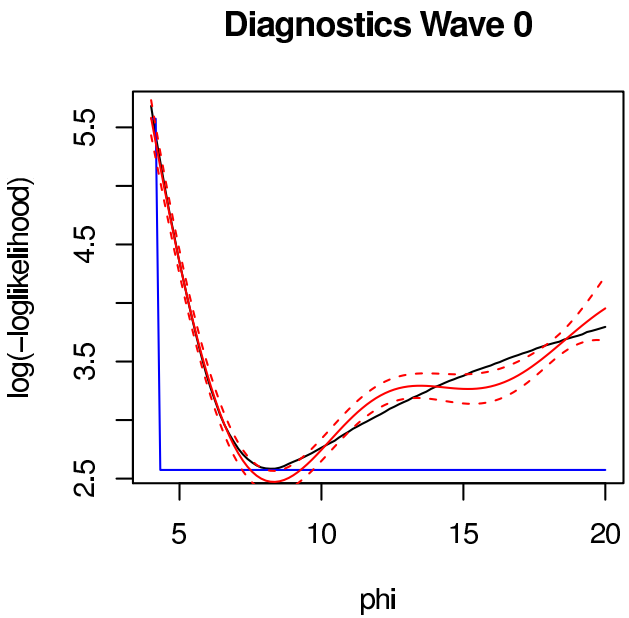




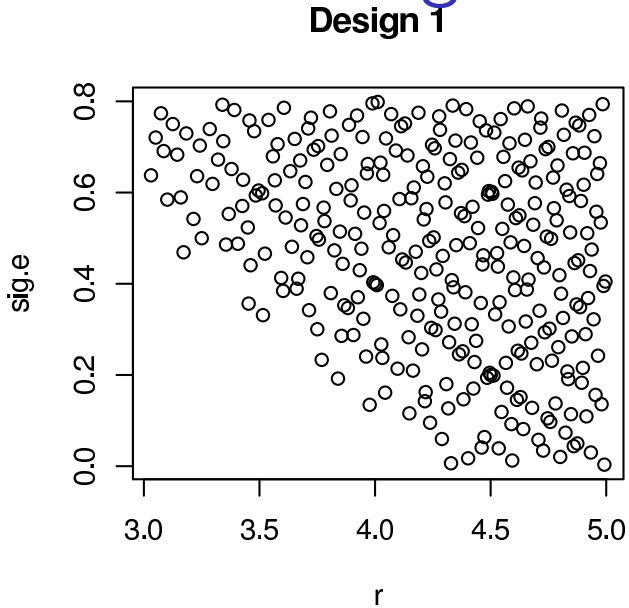
Diagnostics for GP 1 - threshold = 5.6

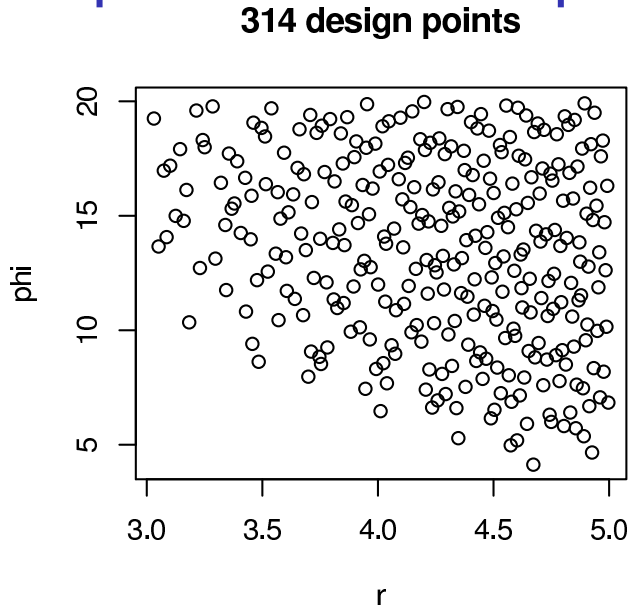


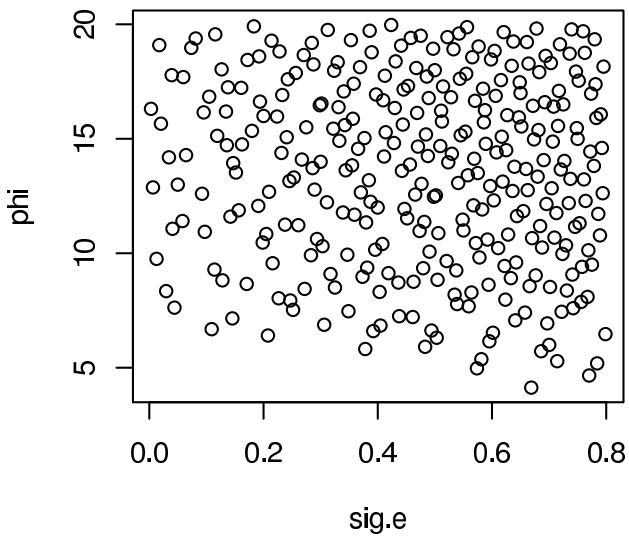




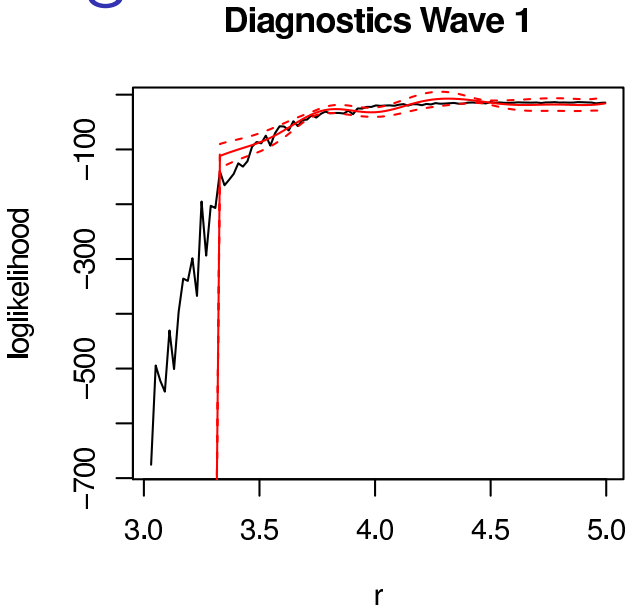
Results - Design 2 - 314 pts - 38% of space implausible

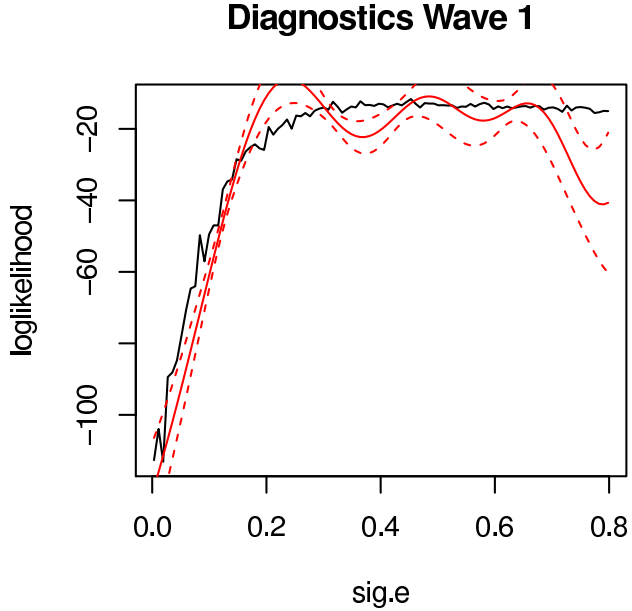


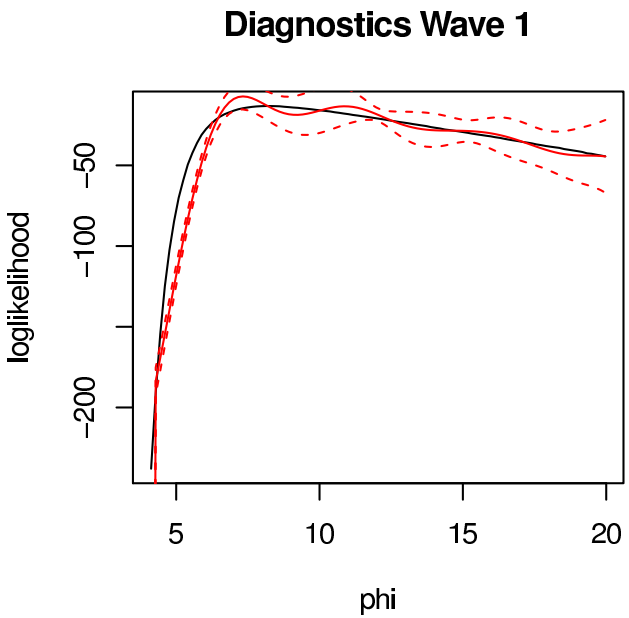




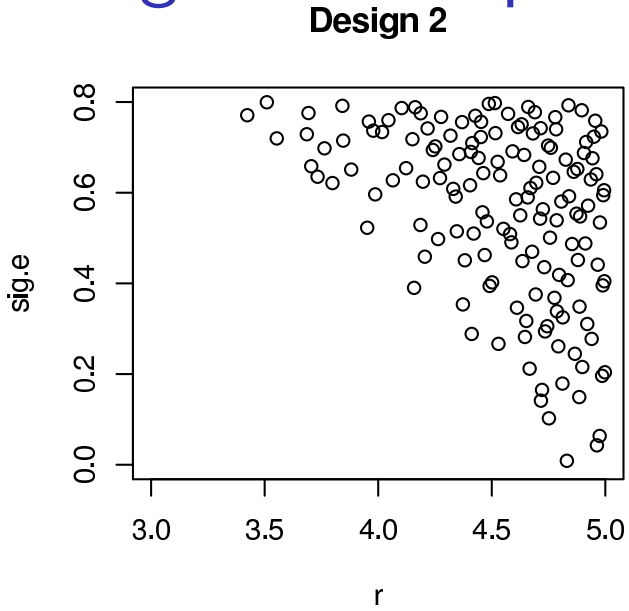
Diagnostics for GP 2 - threshold = -21.8

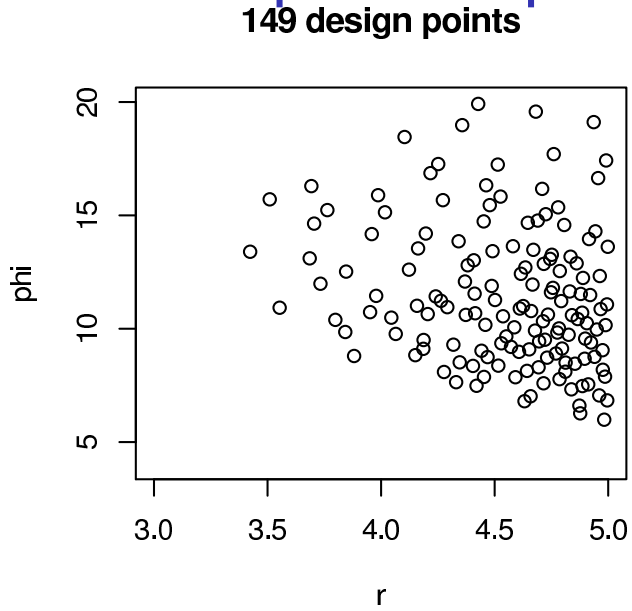


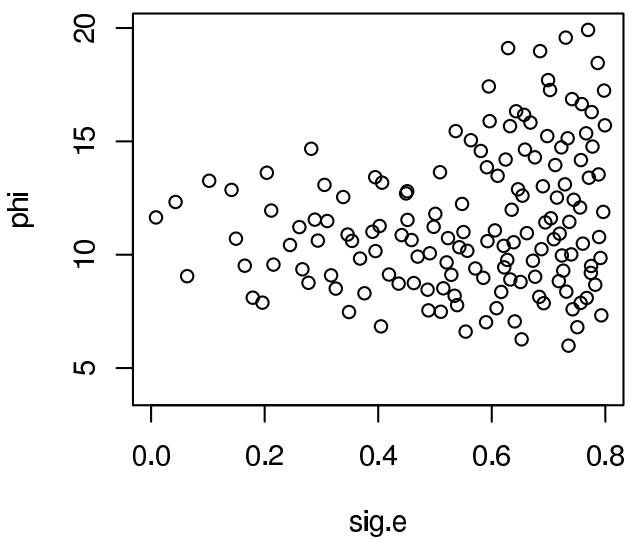




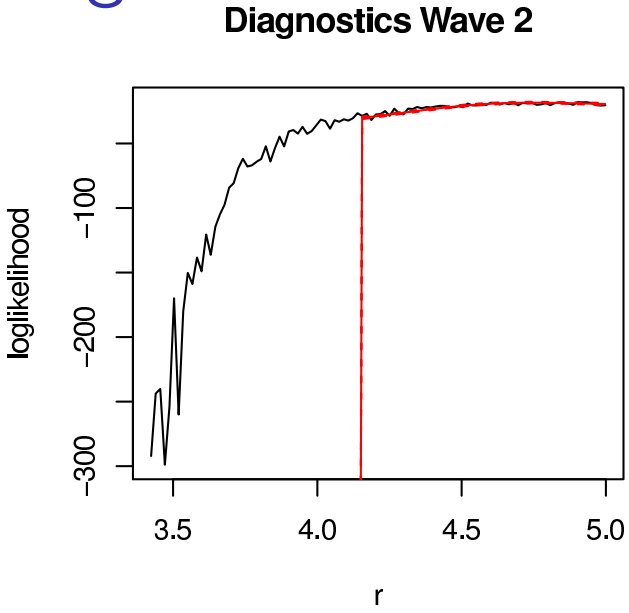
Design 3 - 149 pts - 62% of space implausible

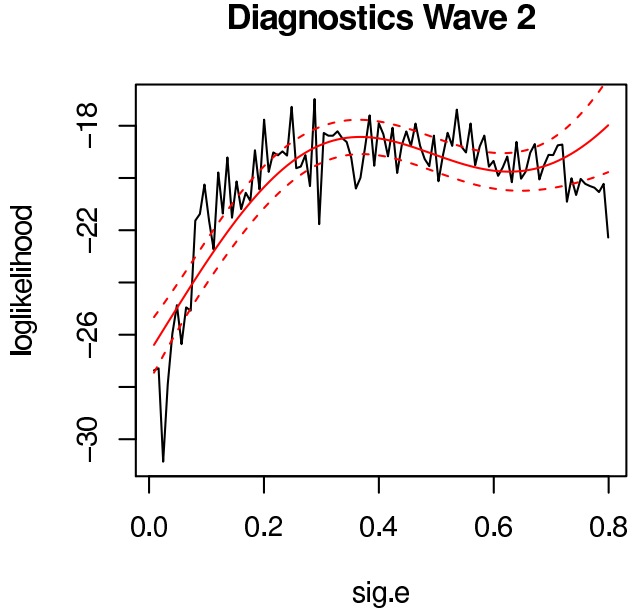


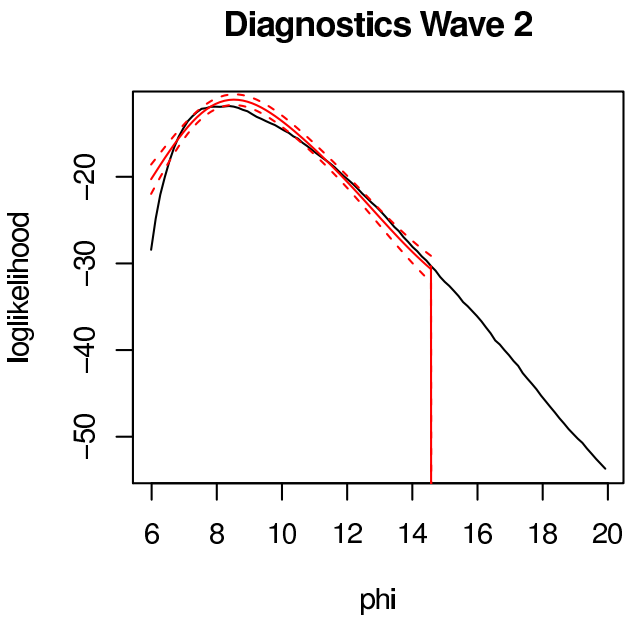




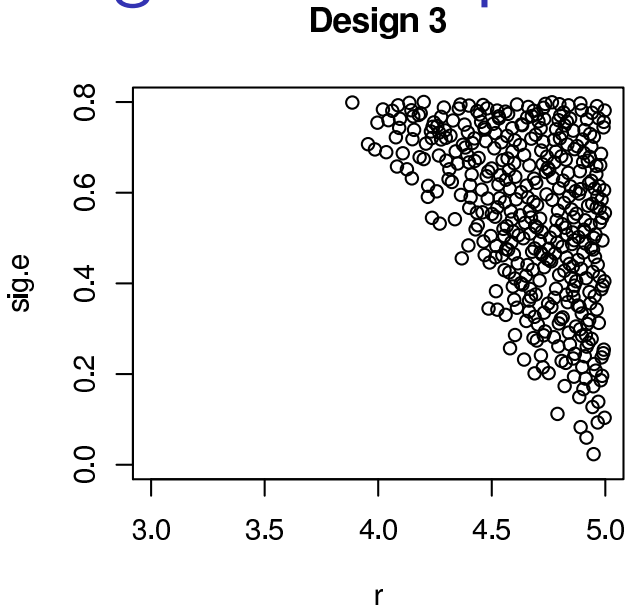
Diagnostics for GP 3 - threshold = -20.7

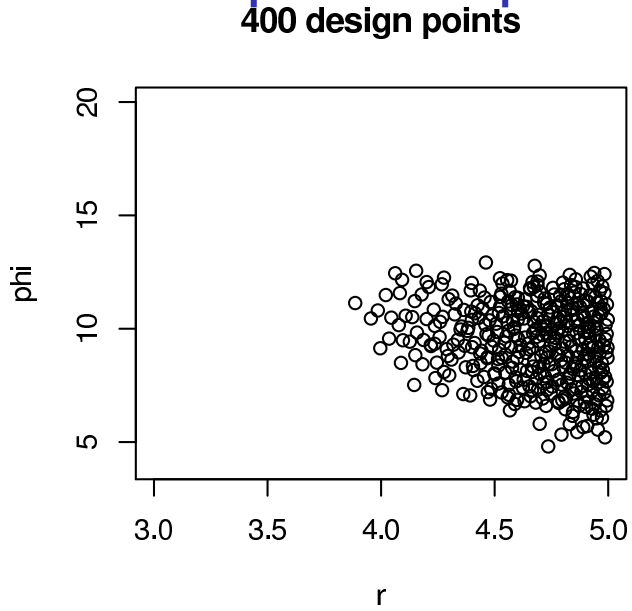


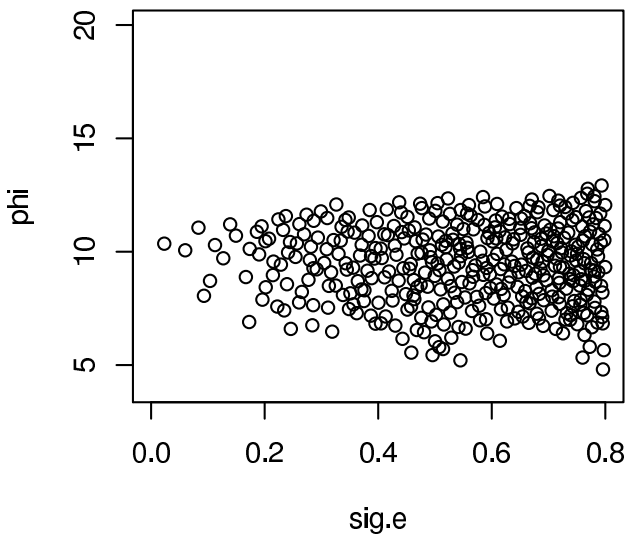




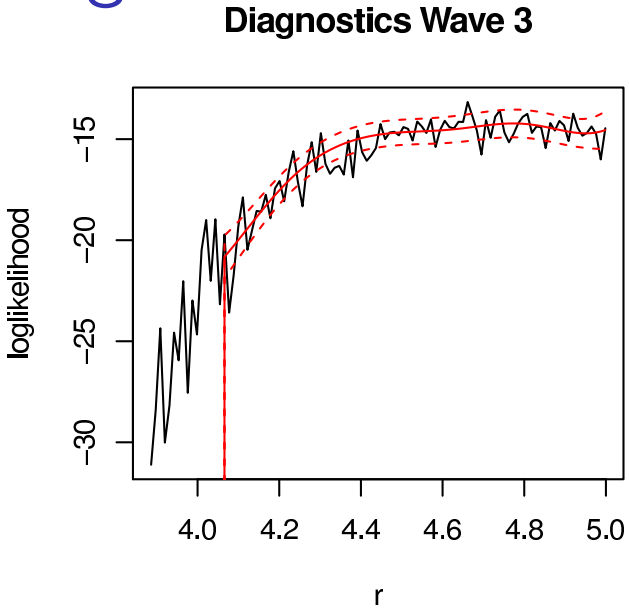
Design 4 - 400 pts - 95% of space implausible

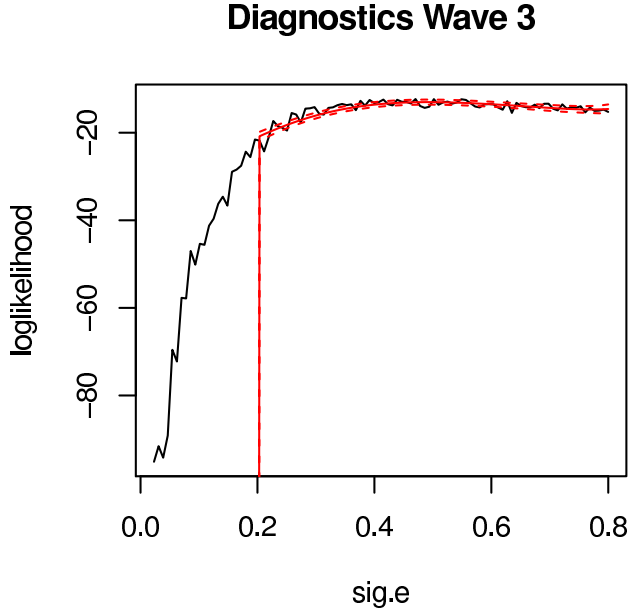


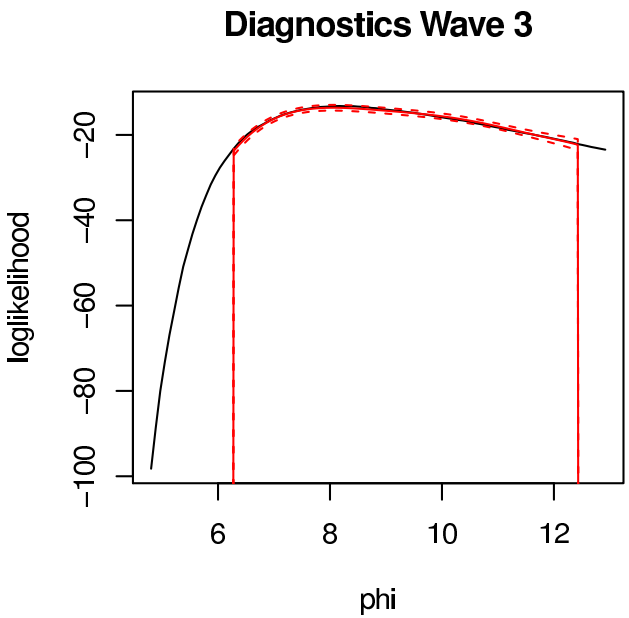




Diagnostics for GP 4 - threshold = -16.4







Finding the posterior

For a given $G(\theta)$, the corresponding posterior is

$$P_G(\theta) = \frac{\Phi(G(\theta))\pi(\theta)}{Z_G} \text{ where } Z_G = \int \Phi(G(\theta))\pi(\theta)\mathrm{d}\theta$$

There are options about what constitutes an answer:

Finding the posterior

For a given $G(\theta)$, the corresponding posterior is

$$P_G(\theta) = \frac{\Phi(G(\theta))\pi(\theta)}{Z_G} \text{ where } Z_G = \int \Phi(G(\theta))\pi(\theta)\mathrm{d}\theta$$

There are options about what constitutes an answer:

- The uncertainty distribution of $P_G(\cdot)$ induced by the uncertainty in G - $\pi(P_G(\cdot)|\mathcal{C})$
  - For a function $h(\theta)$, find the mean and variance of the posterior expectation of $h(\theta)$

$$\mathbb{E}_G\left(\int h(\theta)P_G(\theta)\mathrm{d}\theta \,|\, \mathcal{C}\right) \text{ and } \mathbb{V}\mathrm{ar}_G\left(\int h(\theta)P_G(\theta)\mathrm{d}\theta \,|\, \mathcal{C}\right)$$

Finding the posterior

For a given $G(\theta)$, the corresponding posterior is

$$P_G(\theta) = \frac{\Phi(G(\theta))\pi(\theta)}{Z_G} \text{ where } Z_G = \int \Phi(G(\theta))\pi(\theta)\mathrm{d}\theta$$

There are options about what constitutes an answer:

- The uncertainty distribution of $P_G(\cdot)$ induced by the uncertainty in G - $\pi(P_G(\cdot)|\mathcal{C})$
  - For a function $h(\theta)$, find the mean and variance of the posterior expectation of $h(\theta)$

$$\mathbb{E}_G\left(\int h(\theta)P_G(\theta)\mathrm{d}\theta \,|\, \mathcal{C}\right) \text{ and } \mathbb{V}\mathrm{ar}_G\left(\int h(\theta)P_G(\theta)\mathrm{d}\theta \,|\, \mathcal{C}\right)$$

- Marginal distribution of the posterior:

$$\begin{aligned} P(\theta) = \pi(\theta|\mathcal{D},\mathcal{C}) &= \int \pi(\theta|\mathcal{D},G)\pi(G|\mathcal{C})\mathrm{d}G \\ &= \int \frac{\Phi(G(\theta))\pi(\theta)}{Z_G} \cdot \pi(G|S)\mathrm{d}G \end{aligned}$$

Finding the posterior

Second option $\pi(\theta|\mathcal{D},\mathcal{C})$, seems easier to find and more useful. However, the dependence of the normalising constant Z_G, on G, makes the inference hard.

Finding the posterior

Second option $\pi(\theta|\mathcal{D},\mathcal{C})$, seems easier to find and more useful. However, the dependence of the normalising constant Z_G, on G, makes the inference hard.

An auxiliary variable approach can be used:

- Build a MCMC sampler on

$$p(\theta, G(\theta)|\mathcal{D},\mathcal{C}) = \frac{\Phi(G(\theta))\pi(G(\theta)|\mathcal{C},\theta)\pi(\theta)}{\pi(\mathcal{D}|\mathcal{C})}$$

- The θ-marginal distribution is $p(\theta|\mathcal{D},\mathcal{C})$
- Using MCMC proposal $\theta' \sim q(\theta,\cdot)$, $G'(\theta') \sim \pi(G'(\theta')|\mathcal{C},\theta')$ gives MH acceptance probability

$$\min\left(1, \frac{\Phi(G'(\theta'))\pi(\theta')q(\theta',\theta)}{\Phi(G(\theta))\pi(\theta)q(\theta,\theta')}\right)$$

which is what we may naively have used - propose θ, simulate likelihood value, used standard MCMC.

Note: by building the chain on the extended parameter space $(\theta, G(\theta))$ we avoid having to calculate the normalising constant.

Dealing with GP hyper parameters, Ψ

For $\Psi = (\Sigma, \sigma^2)$, we can

- Estimate and fix Ψ using its MLE (ignoring uncertainty)
- Estimate Ψ and use a Laplace approximation to account for the uncertainty
- Include Ψ in a MCMC scheme and infer its value with the other parameters.

Dealing with GP hyper parameters, Ψ

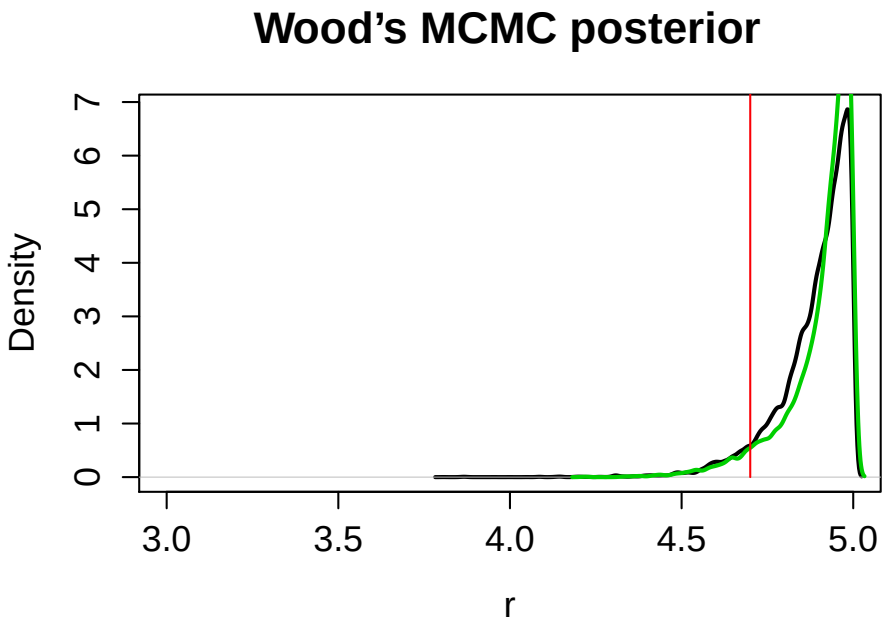
For $\Psi = (\Sigma, \sigma^2)$, we can

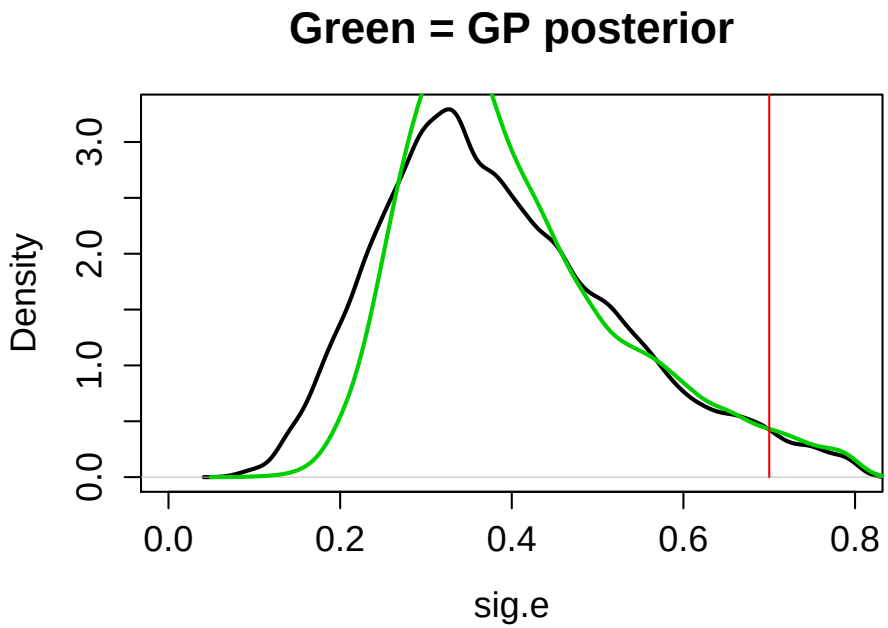
- Estimate and fix Ψ using its MLE (ignoring uncertainty)
- Estimate Ψ and use a Laplace approximation to account for the uncertainty
- Include Ψ in a MCMC scheme and infer its value with the other parameters.

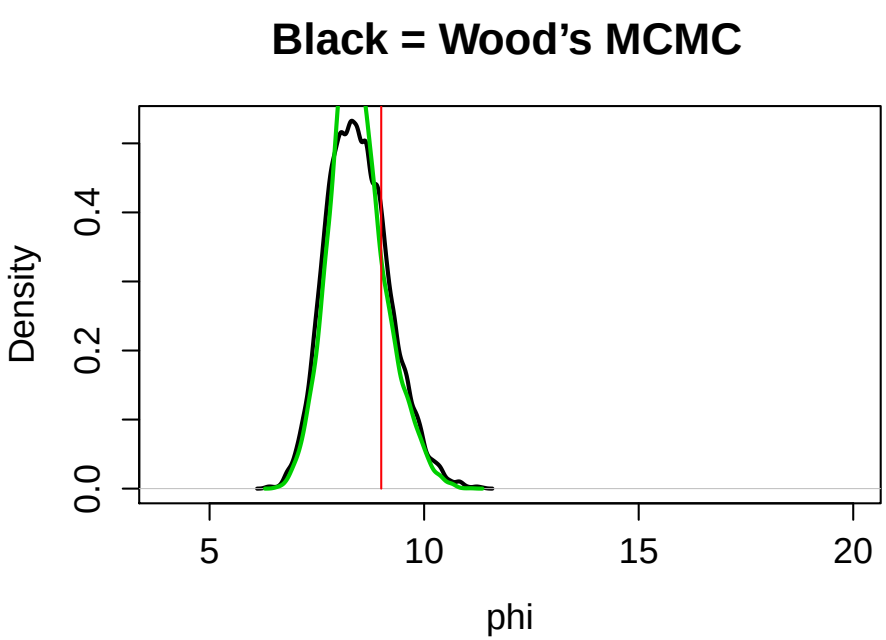
We estimate and fix nugget variance τ from the experimental set up: $\hat{L} = \frac{1}{n} \sum \pi(D|X_i)$

- Variance of $\hat{L}(\theta_i)$ easy to calculate, but $var(\hat{G}(\theta_i))$ is not. We use bootstrapped replicates of the log-likelihood to estimate τ (we could estimate it as part of the GP fitting, but typically this is poorly behaved).

MCMC Results







Computational details

- The Wood MCMC method used $10^5 \times 500$ simulator runs
- The GP code used $(128 + 314 + 149 + 400) = 991 \times 500$ simulator runs
  - 1/100th of the number used by Wood's method.

By the final iteration, the Gaussian processes had ruled out over 98% of the original input space as implausible,

- the MCMC sampler did not need to waste time exploring those regions.

Computational details

- The Wood MCMC method used $10^5 \times 500$ simulator runs
- The GP code used $(128 + 314 + 149 + 400) = 991 \times 500$ simulator runs
  - 1/100th of the number used by Wood's method.

By the final iteration, the Gaussian processes had ruled out over 98% of the original input space as implausible,

- the MCMC sampler did not need to waste time exploring those regions.

Unfortunately though, GPs are computationally expensive to train.

The CPU time taken to run both methods was approximately the same!

For more complex models, there will hopefully be time advantages.

Conclusions

- Monte Carlo methods are naive
  - they don't learn
  - they don't exploit continuity or design considerations

  This makes them powerful, as they will always give the correct answer in time.
- However, computational resource is usually limited.
- If we believe the likelihood is a continuous function of the parameters, and we're prepared to sacrifice asymptotic perfection in the hope of achieving a good approximation in reasonable time, then we can use Gaussian processes to accelerate the inference process.
- Lots still to do
  - justification of threshold values
  - model selection
  - improve MCMC efficiency.
  - ...

Conclusions

- Monte Carlo methods are naive
  - they don't learn
  - they don't exploit continuity or design considerations

  This makes them powerful, as they will always give the correct answer in time.
- However, computational resource is usually limited.
- If we believe the likelihood is a continuous function of the parameters, and we're prepared to sacrifice asymptotic perfection in the hope of achieving a good approximation in reasonable time, then we can use Gaussian processes to accelerate the inference process.
- Lots still to do
  - justification of threshold values
  - model selection
  - improve MCMC efficiency.
  - ...

Thank you for listening!