Using surrogate models to accelerate parameter estimation for complex simulators

Richard Wilkinson, James Hensman

University of Sheffield
Lancaster University

Talk plan

(a) Emulation

(b) Calibration - history matching and ABC

(c) GP-ABC

   - Design

Talk plan

(a) Emulation

(b) Calibration - history matching and ABC

(c) GP-ABC

- Design

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 estimate tunable parameters?
- how do we deal with computational constraints?

Surrogate/Meta-modelling Emulation

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

- All inference must be done using a finite ensemble of model runs

$$\mathcal{D}_{sim} = \{(\theta_i, f(\theta_i))\}_{i=1,\dots,N}$$

- If θ is not in the ensemble, then we are uncertain about the value of $f(\theta)$.

Code uncertainty

For complex simulators, run times might be long, ruling out brute-force approaches such as Monte Carlo methods.

- All inference must be done using a finite ensemble of model runs

$$\mathcal{D}_{sim} = \{(\theta_i, f(\theta_i))\}_{i=1,\dots,N}$$

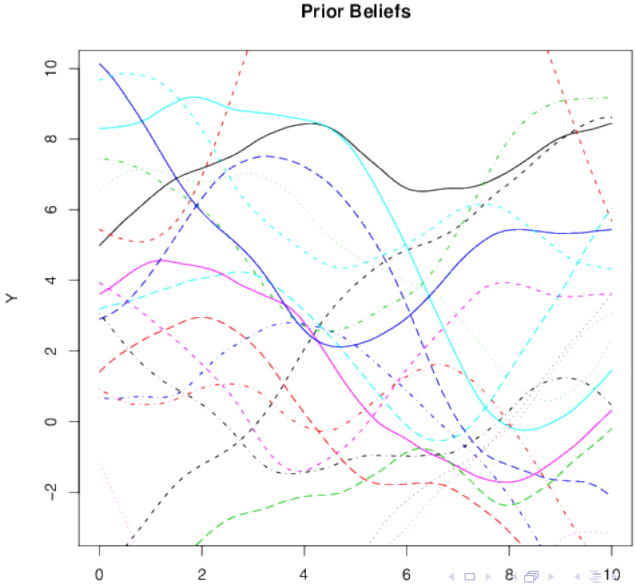
- If θ is not in the ensemble, then we are uncertain about the value of $f(\theta)$.

Idea: If the simulator is expensive, build a cheap model (*surrogate or emulator*) of it and use this in any analysis.

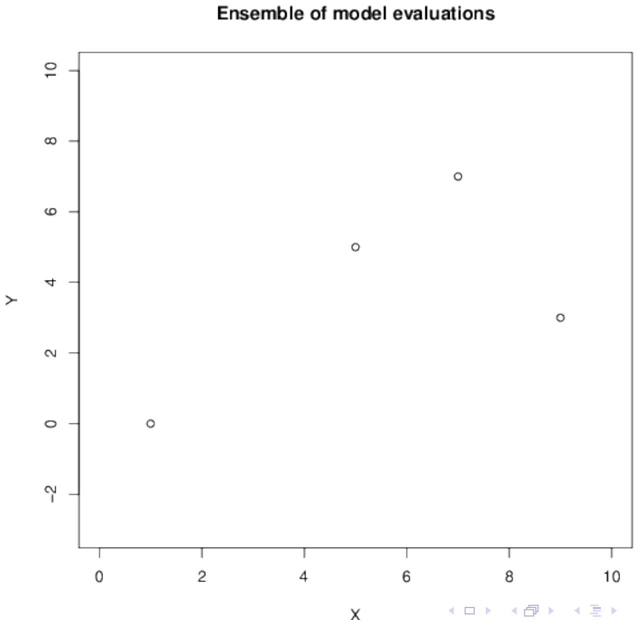
'a model of the model'

Gaussian Process Illustration

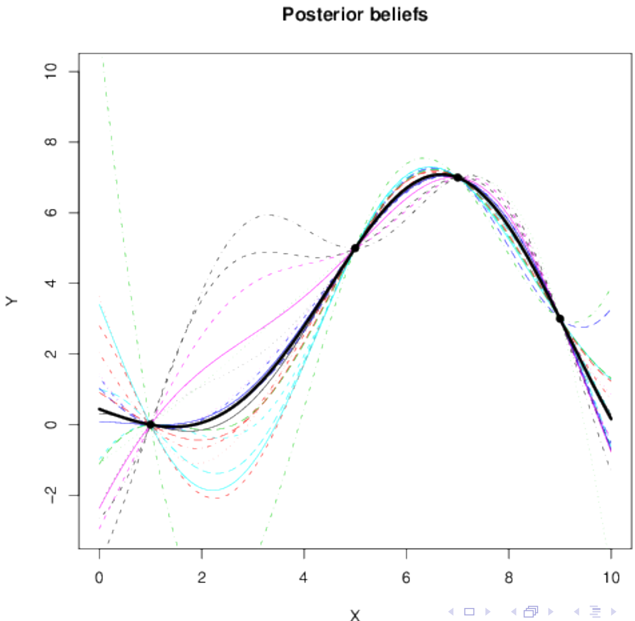
Zero mean



Gaussian Process Illustration



Gaussian Process Illustration



Storage Panacea

Knowledge of the physical problem is encoded in a simulator f

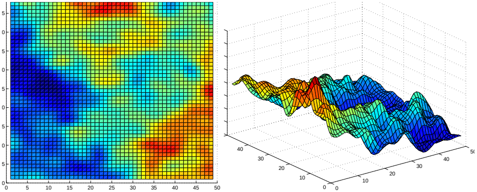
Inputs:

Permeability field, K
(2d field)

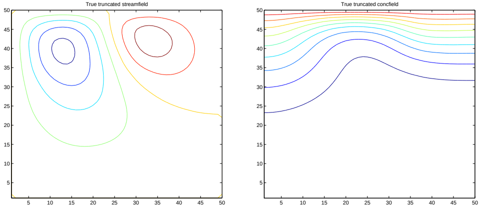
$$\downarrow f(K) \downarrow$$

Outputs:

Stream func. (2d field),
concentration (2d field),
surface flux (1d scalar),
$\vdots$



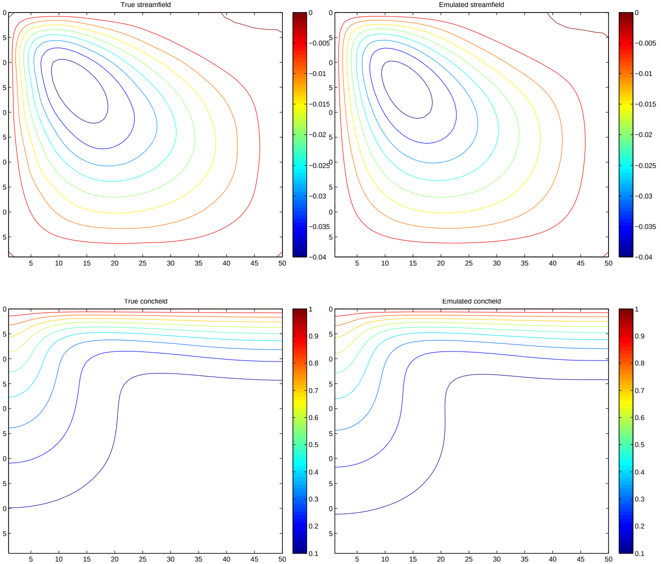
$\downarrow f(K)$



Surface Flux$= 6.43, \ldots$

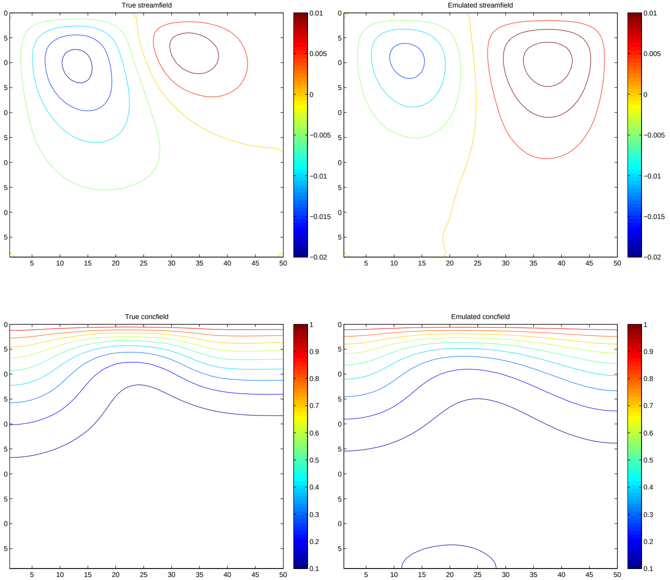
CCS examples

Left=true, right = emulated, 118 training runs, held out test set.



Emulating the stream function and concentration fields

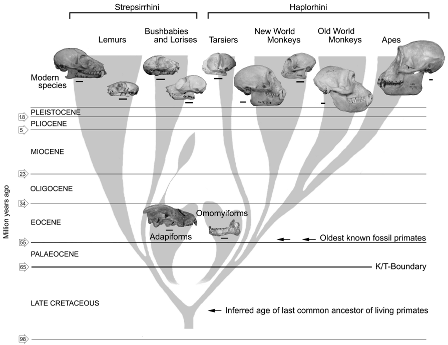
Left=true, right = emulated, 118 training runs, held out test set.

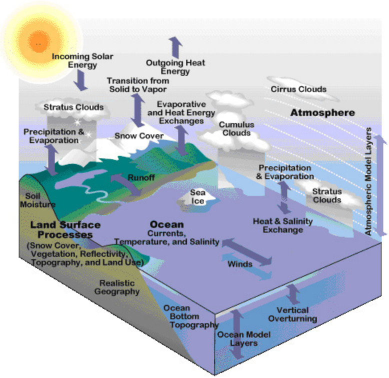


Calibration: history matching and ABC

Inverse problems

- For most simulators we specify parameters θ and i.c.s and the simulator, $f(\theta)$, generates output X.
- The inverse-problem: observe data D, estimate parameter values θ





Two approaches

Probabilistic calibration

Find the posterior distribution

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

for likelihood function
$\pi(\mathcal{D}|\theta) = \int \pi(D|X,\theta)\pi(X|\theta)\mathrm{d}X$
which relates the simulator output, to the data,e.g.,

$$D = X + e + \epsilon$$

where $e \sim N(0, \sigma_e^2)$ represents simulator discrepancy, and $\epsilon \sim N(0, \sigma_\epsilon^2)$ represents measurement error on the data

Two approaches

Probabilistic calibration

Find the posterior distribution

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

for likelihood function $\pi(\mathcal{D}|\theta) = \int \pi(D|X,\theta)\pi(X|\theta)\mathrm{d}X$ which relates the simulator output, to the data,e.g.,

$$D = X + e + \epsilon$$

where $e \sim N(0, \sigma_e^2)$ represents simulator discrepancy, and $\epsilon \sim N(0, \sigma_\epsilon^2)$ represents measurement error on the data

History matching

Find the plausible parameter set

$$\mathcal{P}_\theta = \{\theta : f(\theta) \in \mathcal{P}_D\}$$

where $\mathcal{P}_D$ is some plausible set of simulation outcomes that are consistent with simulator discrepancy and measurement error, e.g.,

$$\mathcal{P}_D = \{X : |D - X| \leq 3(\sigma_e + \sigma_\epsilon)\}$$

Two approaches

Probabilistic calibration

Find the posterior distribution

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

for likelihood function $\pi(\mathcal{D}|\theta) = \int \pi(D|X,\theta)\pi(X|\theta)\mathrm{d}X$ which relates the simulator output, to the data,e.g.,

$$D = X + e + \epsilon$$

where $e \sim N(0, \sigma_e^2)$ represents simulator discrepancy, and $\epsilon \sim N(0, \sigma_\epsilon^2)$ represents measurement error on the data

History matching

Find the plausible parameter set

$$\mathcal{P}_\theta = \{\theta : f(\theta) \in \mathcal{P}_D\}$$

where $\mathcal{P}_D$ is some plausible set of simulation outcomes that are consistent with simulator discrepancy and measurement error, e.g.,

$$\mathcal{P}_D = \{X : |D - X| \leq 3(\sigma_e + \sigma_\epsilon)\}$$

Calibration finds a distribution representing plausible parameter values;
History matching classifies parameter space as plausible or implausible.

Calibration - 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 are popular in biological disciplines, particularly genetics. They are

- Simple to implement
- Intuitive
- Embarrassingly parallelizable
- Can usually be applied

Rejection ABC

Uniform Rejection Algorithm

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

Rejection ABC

Uniform Rejection Algorithm

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

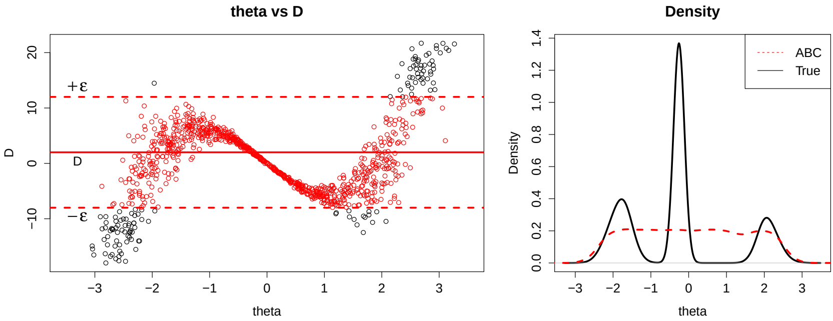
ϵ reflects the tension between computability and accuracy.

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

Rejection sampling is inefficient, but we can adapt other MC samplers such as MCMC and SMC.

Simple → Popular with non-statisticians

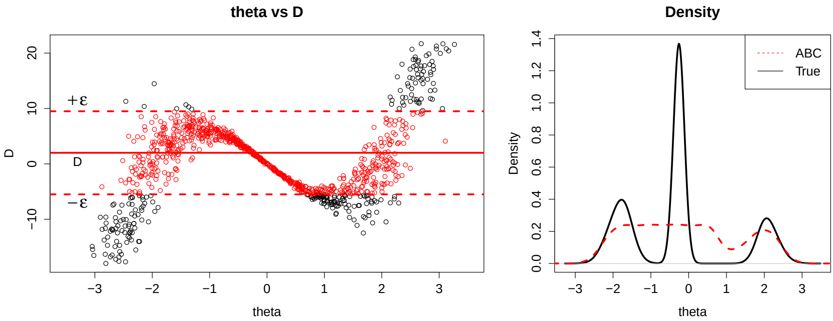
$\epsilon = 10$



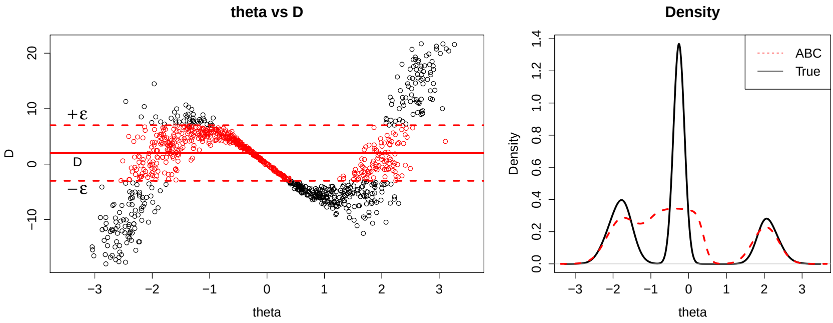
$$\theta \sim U[-10, 10], \qquad X \sim N(2(\theta + 2)\theta(\theta - 2), 0.1 + \theta^2)$$

$$\rho(D, X) = |D - X|, \qquad D = 2$$

$\epsilon = 7.5$

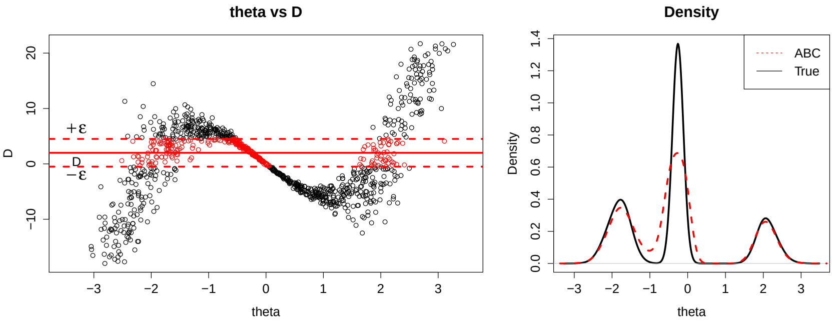

theta vs D
20
10
0
−10
D
+ε
D
−ε
−3
−2
−1
0
1
2
3
theta
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Density
ABC
True
−3
−2
−1
0
1
2
3
theta

$\epsilon = 5$

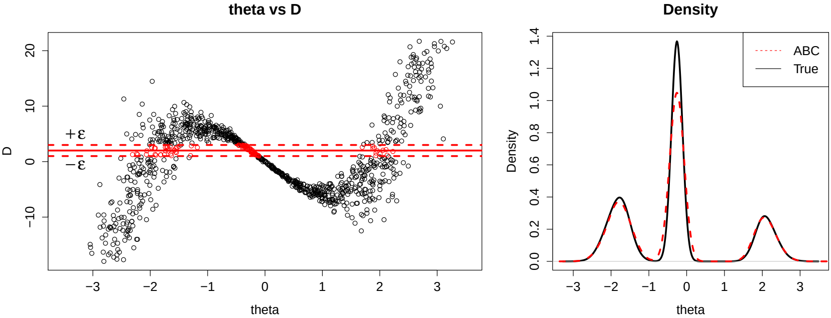


theta vs D
D
20
10
0
−10
+ε
D
−ε
−3
−2
−1
0
1
2
3
theta
Density
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
ABC
True
−3
−2
−1
0
1
2
3
theta

$\epsilon = 2.5$


theta vs D
+ε
D
−ε
20
10
0
−10
D
−3
−2
−1
0
1
2
3
theta
Density
ABC
True
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Density
−3
−2
−1
0
1
2
3
theta

$\epsilon = 1$


theta vs D
D
20
10
0
−10
+ε
−ε
−3
−2
−1
0
1
2
3
theta
Density
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
ABC
True
−3
−2
−1
0
1
2
3
theta

Limitations of Monte Carlo methods

(Non approximate) Monte Carlo methods are generally guaranteed to succeed if we run them for long enough, but can require more simulation than is possible.

Limitations of Monte Carlo methods

(Non approximate) Monte Carlo methods are generally guaranteed to succeed if we run them for long enough, but can require more simulation than is possible.

Most MC methods

- sample naively - they don't learn from previous simulations.
- don't exploit known properties of the likelihood function, such as continuity
- sample randomly, rather than using careful design.

> *"Whenever there is a randomised way of doing something, there is a non-randomised way which yields better results, but requires more thinking" Jaynes*

Limitations of Monte Carlo methods

(Non approximate) Monte Carlo methods are generally guaranteed to succeed if we run them for long enough, but can require more simulation than is possible.

Most MC methods

- sample naively - they don't learn from previous simulations.
- don't exploit known properties of the likelihood function, such as continuity
- sample randomly, rather than using careful design.

> *"Whenever there is a randomised way of doing something, there is a non-randomised way which yields better results, but requires more thinking" Jaynes*

Using surrogate models we can avoid some of these 'weaknesses'.

Target of approximation

What should we approximate with the surrogate model?

- simulator output

- Likelihood function

Target of approximation

What should we approximate with the surrogate model?

- simulator output
  - often easy to work with
  - often high dimensional
  - requires a global approximation, i.e., need to predict $f(\theta)$ at all θ of interest.
  - if the simulator is stochastic, the distribution of $f(\theta)$ at fixed θ is often not Gaussian.
- Likelihood function

Target of approximation

What should we approximate with the surrogate model?

- simulator output
  - often easy to work with
  - often high dimensional
  - requires a global approximation, i.e., need to predict $f(\theta)$ at all θ of interest.
  - if the simulator is stochastic, the distribution of $f(\theta)$ at fixed θ is often not Gaussian.
- Likelihood function
  - 1 dimensional surface
  - allows us to focus on the data, i.e., predict $\log L(\theta|D_{obs})$ at all θ. The data D_{obs} is fixed
  - hard to model
  - hard to gain physical insights - primarily useful for calibration

Likelihood estimation

Wilkinson 2013

It can be shown that ABC replaces the true likelihood $\pi(D|\theta)$ by an ABC likelihood

$$\pi_{ABC}(D|\theta) = \int \pi_\epsilon(D|X)\pi(X|\theta)\mathrm{dX}$$

where $\pi_\epsilon(D|X)$ is the ABC acceptance kernel (often $\mathbb{I}_{\rho(D,X)<\epsilon}$)

Likelihood estimation

Wilkinson 2013

It can be shown that ABC replaces the true likelihood $\pi(D|\theta)$ by an ABC likelihood

$$\pi_{ABC}(D|\theta) = \int \pi_\epsilon(D|X)\pi(X|\theta)\mathrm{dX}$$

where $\pi_\epsilon(D|X)$ is the ABC acceptance kernel (often $\mathbb{I}_{\rho(D,X)<\epsilon}$)

We can estimate this using repeated runs from the simulator

$$\hat{\pi}_{ABC}(D|\theta) \approx \frac{1}{N}\sum \pi_\epsilon(D|X_i)$$

where $X_i \sim \pi(X|\theta)$.

Likelihood estimation

Wilkinson 2013

It can be shown that ABC replaces the true likelihood $\pi(D|\theta)$ by an ABC likelihood

$$\pi_{ABC}(D|\theta) = \int \pi_\epsilon(D|X)\pi(X|\theta)\mathrm{dX}$$

where $\pi_\epsilon(D|X)$ is the ABC acceptance kernel (often $\mathbb{I}_{\rho(D,X)<\epsilon}$)

We can estimate this using repeated runs from the simulator

$$\hat{\pi}_{ABC}(D|\theta) \approx \frac{1}{N}\sum \pi_\epsilon(D|X_i)$$

where $X_i \sim \pi(X|\theta)$.

We can model $\log L(\theta) = \log \pi_{ABC}(D|\theta)$ and use this to find the posterior.

Requires the likelihood to be continuous and smooth

History matching waves

Wilkinson 2014

The log-likelihood $l(\theta) = \log L(\theta)$ typical ranges across too a range of values, consequently, most models struggle to accurately approximate the log-likelihood across the entire parameter space.

History matching waves

Wilkinson 2014

The log-likelihood $l(\theta) = \log L(\theta)$ typical ranges across too a range of values, consequently, most models struggle to accurately approximate the log-likelihood across the entire parameter space.

- Introduce waves of history matching.
- In each wave, build a GP model that can rule out regions of space as implausible.

History matching waves

Wilkinson 2014

The log-likelihood $l(\theta) = \log L(\theta)$ typical ranges across too a range of values, consequently, most models struggle to accurately approximate the log-likelihood across the entire parameter space.

- Introduce waves of history matching.
- In each wave, build a GP model that can rule out regions of space as implausible.

We decide that θ is implausible if

$$\mathbb{P}(\tilde{l}(\theta) > \max_{\theta_i} l(\theta_i) - T) \leq 0.001$$

where $\tilde{l}(\theta)$ is the GP model of $\log \pi(D|\theta)$

Choose T so that if $l(\hat{\theta}) - l(\theta) > T$ then $\pi(\theta|y) \approx 0$.

- Ruling θ to be implausible is to set $\pi(\theta|y) = 0$

The choice of T is problem specific, and we often start with a large T to ensure a conservative criterion.

Example: Ricker Model

Wood 2010

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

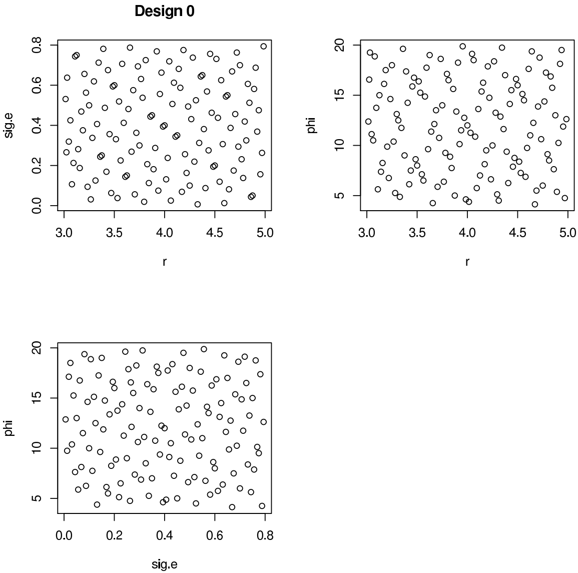
- Let N_t denote the number of animals at time t.

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

  where e_t are independent $N(0, \sigma_e^2)$ process noise
- Assume we observe counts y_t where

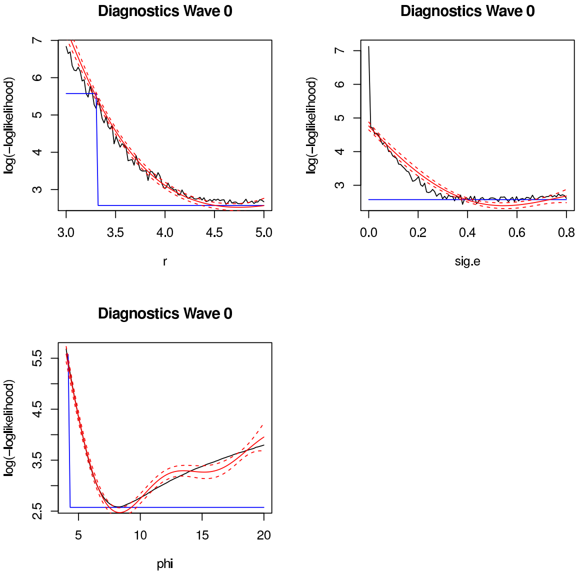
$$y_t \sim Po(\phi N_t)$$

Results - Design 1 - 128 pts

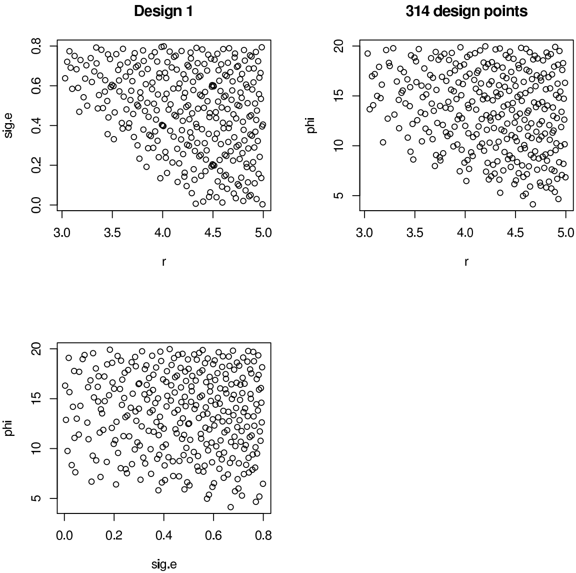


Diagnostics for GP 1 modelling $\log(-\log l(\theta))$

Threshold $= 5.6$

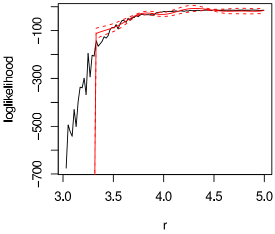


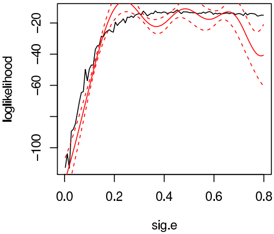
Results - Design 2 - 314 pts - 38% of space implausible

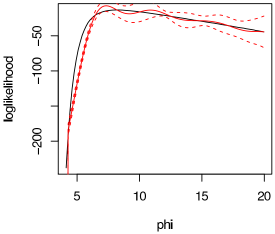


Diagnostics for GP 2 modelling log $l(\theta)$

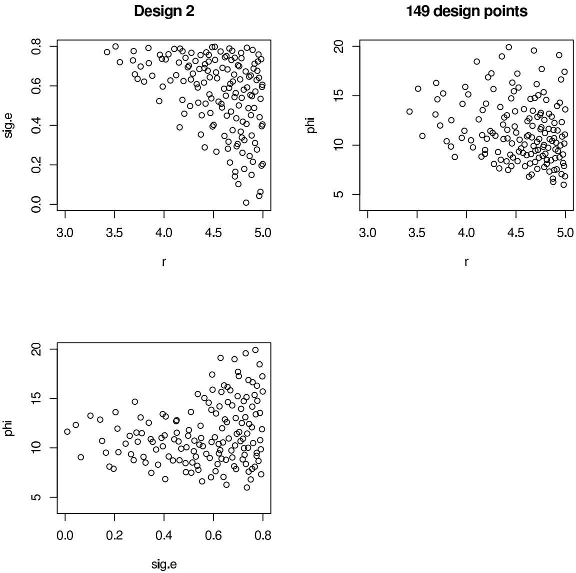
threshold = -21.8


Diagnostics Wave 1


Diagnostics Wave 1

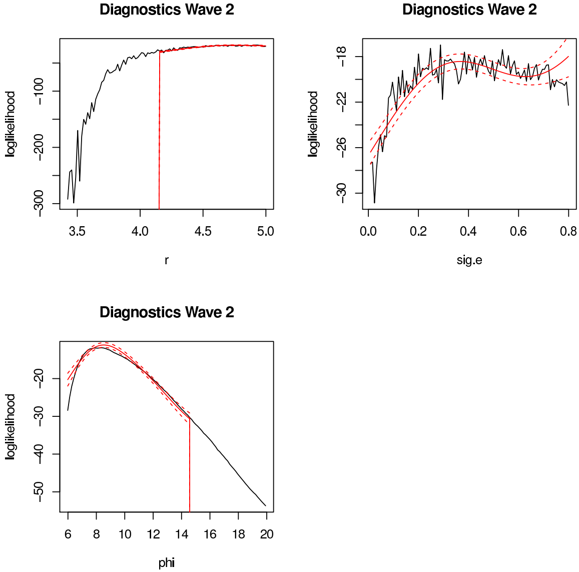

Diagnostics Wave 1

Design 3 - 149 pts - 62% of space implausible

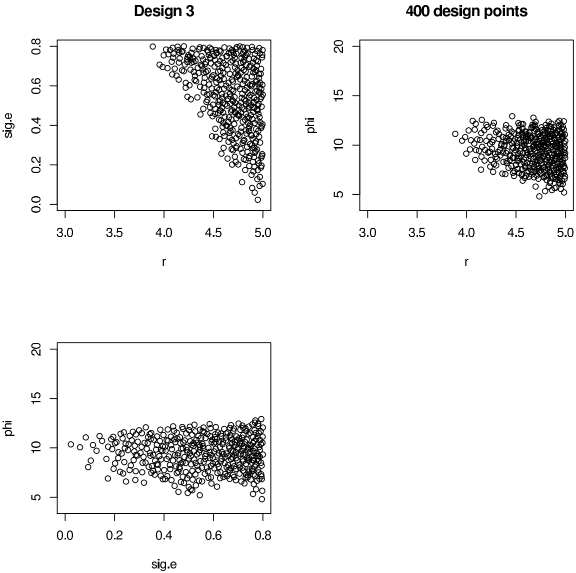


Diagnostics for GP 3 modelling log $l(\theta)$

Threshold = -20.7

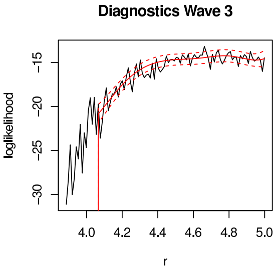


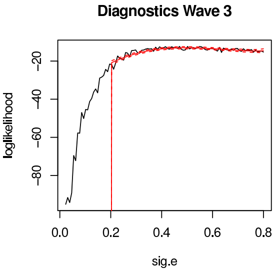
Design 4 - 400 pts - 95% of space implausible

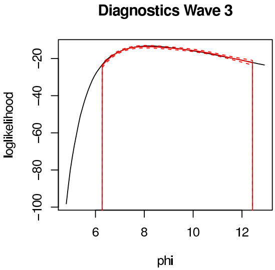


Diagnostics for GP 4 modelling $\log l(\theta)$

Threshold = -16.4

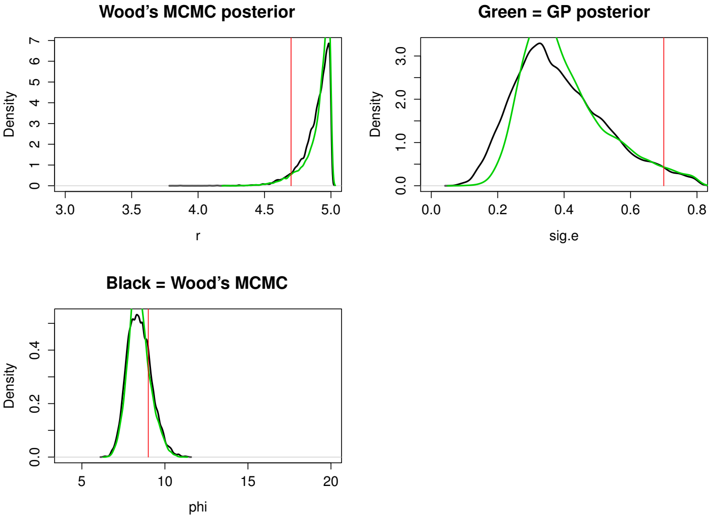

Diagnostics Wave 3


Diagnostics Wave 3


Diagnostics Wave 3

MCMC Results

Comparison with Wood 2010, synthetic likelihood approach



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.

Design for calibration

with James Hensman

Implausibility

When using emulators for history-matching and ABC, the aim is to accurately classify space as plausible or implausible by estimating the probability

$$p(\theta) = \mathbb{P}(\theta \in \mathcal{P}_\theta)$$

where $\mathcal{P}_\theta = \{\theta : f(\theta) \in \mathcal{P}_D\}$, based upon a GP model of the simulator or likelihood

$$f(\theta) \sim GP(m(\cdot), c(\cdot, \cdot))$$

Implausibility

When using emulators for history-matching and ABC, the aim is to accurately classify space as plausible or implausible by estimating the probability

$$p(\theta) = \mathbb{P}(\theta \in \mathcal{P}_\theta)$$

where $\mathcal{P}_\theta = \{\theta : f(\theta) \in \mathcal{P}_D\}$, based upon a GP model of the simulator or likelihood

$$f(\theta) \sim GP(m(\cdot), c(\cdot, \cdot))$$

The key determinant of emulator accuracy is the design used to train the GP

$$D_n = \{\theta_i, f(\theta_i)\}_{i=1}^N$$

Usual design choices are space filling designs

- Maximin latin hypercubes, Sobol sequences

Implausibility

When using emulators for history-matching and ABC, the aim is to accurately classify space as plausible or implausible by estimating the probability

$$p(\theta) = \mathbb{P}(\theta \in \mathcal{P}_\theta)$$

where $\mathcal{P}_\theta = \{\theta : f(\theta) \in \mathcal{P}_D\}$, based upon a GP model of the simulator or likelihood

$$f(\theta) \sim GP(m(\cdot), c(\cdot, \cdot))$$

The key determinant of emulator accuracy is the design used to train the GP

$$D_n = \{\theta_i, f(\theta_i)\}_{i=1}^N$$

Usual design choices are space filling designs

- Maximin latin hypercubes, Sobol sequences

Calibration doesn't need a global approximation to the simulator - this is wasteful

Entropic designs

Instead build a sequential design $\theta_1, \theta_2, \ldots$ using the current classification

$$p(\theta) = \mathbb{P}(\theta \in \mathcal{P}_\theta | D_n)$$

to guide the choice of design points

Entropic designs

Instead build a sequential design $\theta_1, \theta_2, \dots$ using the current classification

$$p(\theta) = \mathbb{P}(\theta \in \mathcal{P}_\theta | D_n)$$

to guide the choice of design points
First idea: add design points where we are most uncertain

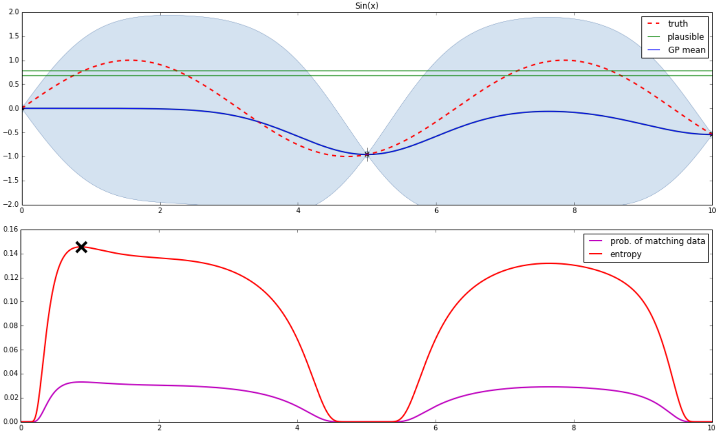
- The entropy of the classification surface is

$$E(\theta) = -p(\theta) \log p(\theta) - (1 - p(\theta)) \log(1 - p(\theta))$$

- Choose the next design point where we are most uncertain.

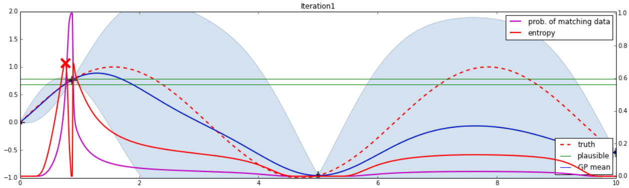
$$\theta_{n+1} = \arg\max E(\theta)$$

Toy 1d example $f(\theta) = \sin\theta$

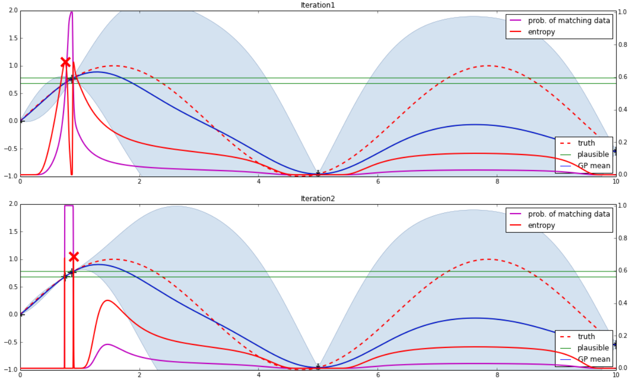


Add a new design point (simulator evaluation) at the point of greatest entropy

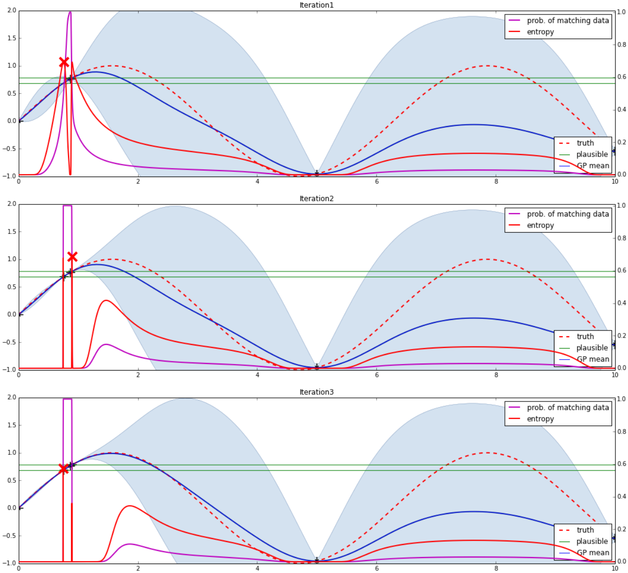
Toy 1d example $f(\theta) = \sin\theta$



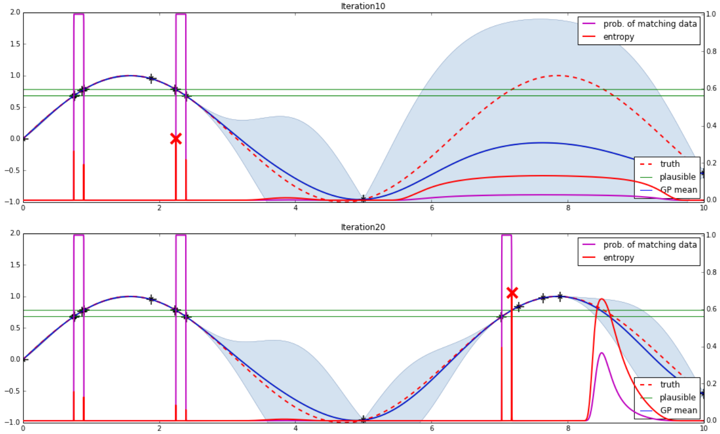
Toy 1d example $f(\theta) = \sin\theta$



Toy 1d example $f(\theta) = \sin\theta$



Toy 1d example $f(\theta) = \sin\theta$ - After 10 and 20 iterations



This criterion spends too long resolving points at the edge of the classification region.

- not enough exploration

Expected average entropy

Chevalier *et al.* 2014

Instead, we can find the average entropy of the classification surface

$$E_n = \int E(\theta)\mathrm{d}\theta$$

where n denotes it is based on the current design of size n.

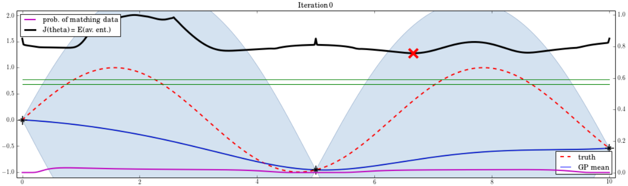
- Choose the next design point, θ_{n+1}, to minimise the expected average entropy

  $$\theta_{n+1} = \arg\min J_n(\theta)$$

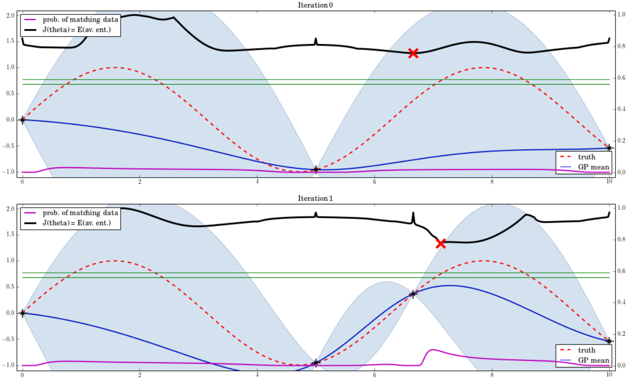
  where

  $$J_n(\theta) = \mathbb{E}(E_{n+1}|\theta_{n+1} = \theta)$$

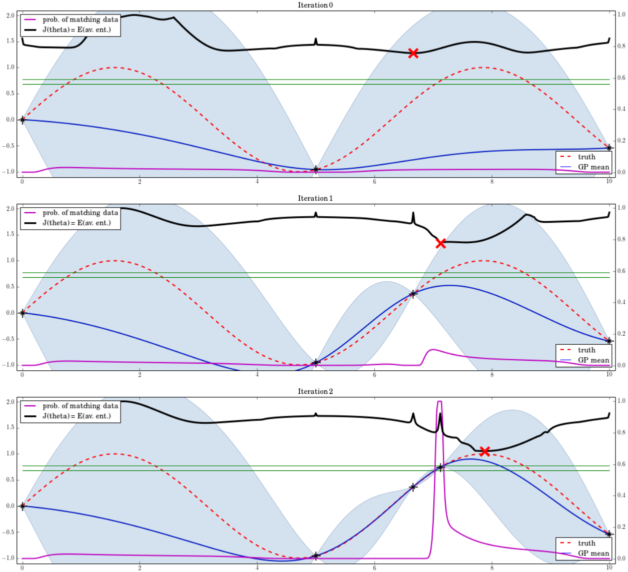
Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



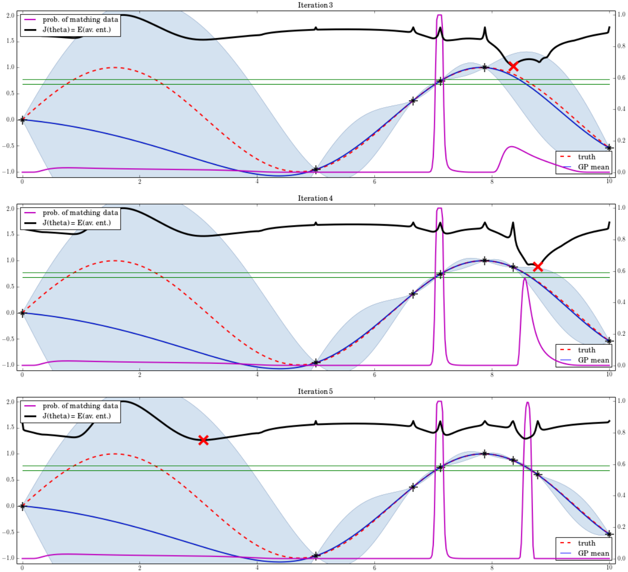
Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy

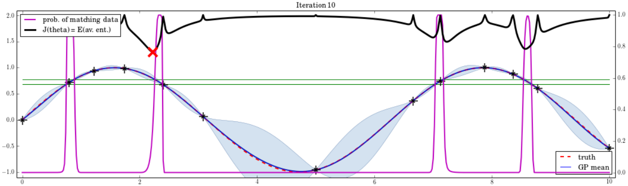


Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



Toy 1d: min expected entropy vs max entropy

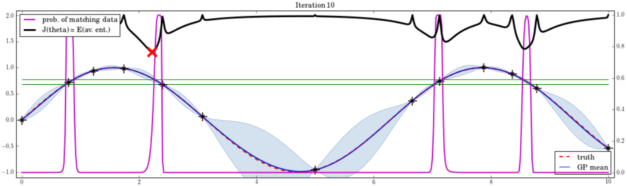
After 10 iterations, choosing the point of maximum entropy



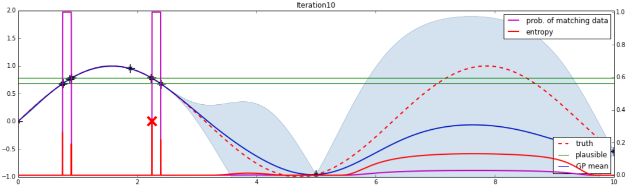
we have found the plausible region to reasonable accuracy.

Toy 1d: min expected entropy vs max entropy

After 10 iterations, choosing the point of maximum entropy



we have found the plausible region to reasonable accuracy. Whereas maximizing the entropy has not



In 1d, a simpler space filling criterion would work just as well.

Solving the optimisation problem

Finding θ which minimises $J_n(\theta) = \mathbb{E}(E_{n+1}|\theta_{n+1} = \theta)$ is expensive.

- Even for 3d problems, grid search is prohibitively expensive
- Dynamic grids help

Solving the optimisation problem

Finding θ which minimises $J_n(\theta) = \mathbb{E}(E_{n+1}|\theta_{n+1} = \theta)$ is expensive.

- Even for 3d problems, grid search is prohibitively expensive
- Dynamic grids help

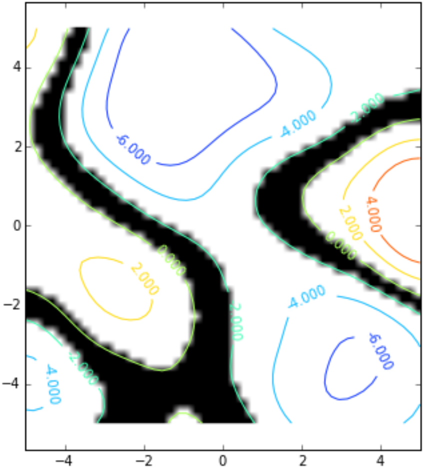
We can use Bayesian optimization to find the optima:

1. Evaluate $J_n(\theta)$ at a small number of locations
2. Build a GP model of $J_n(\cdot)$
3. Choose the next θ at which to evaluate J_n so as to minimise the expected-improvement (EI) criterion
4. Return to step 2.

History match

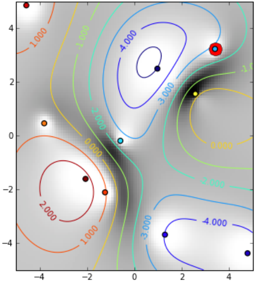
Can we learn the following plausible set?

- A sample from a GP on $\mathbb{R}^2$.
- Find x s.t. $-2 < f(x) < 0$



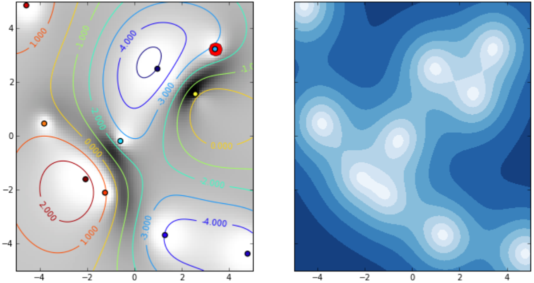
Iteration 10

Left=$p(\theta)$, middle= $E(\theta)$, right = $\tilde{J}(\theta)$



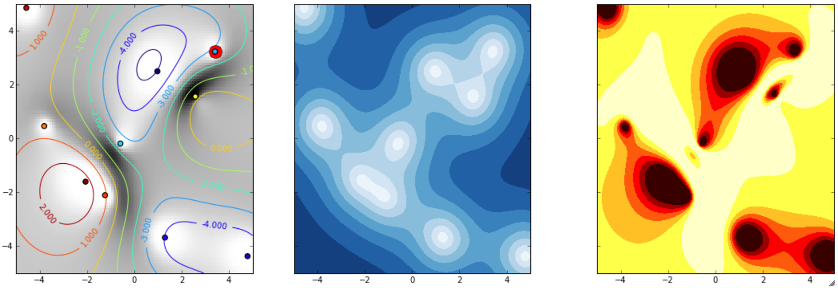
Iteration 10

Left$=p(\theta)$, middle$= E(\theta)$, right $= \tilde{J}(\theta)$



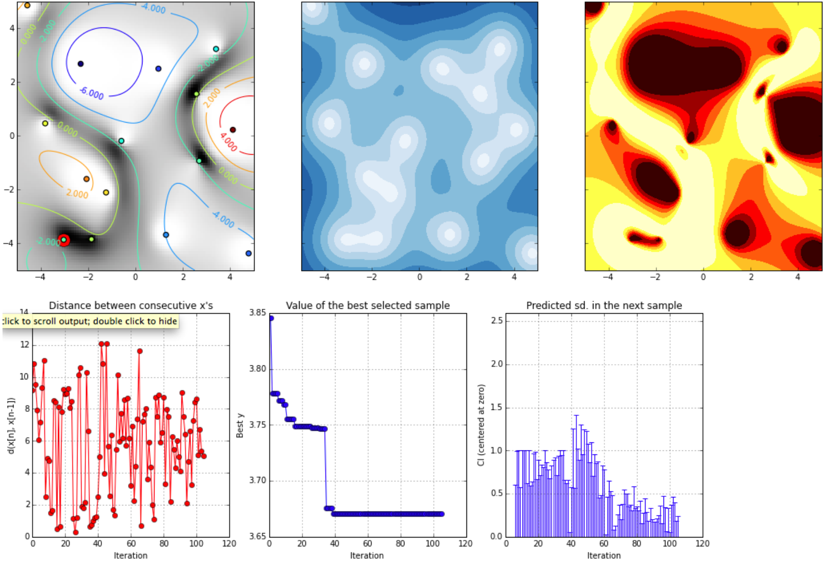
Iteration 10

Left=$p(\theta)$, middle= $E(\theta)$, right = $\tilde{J}(\theta)$

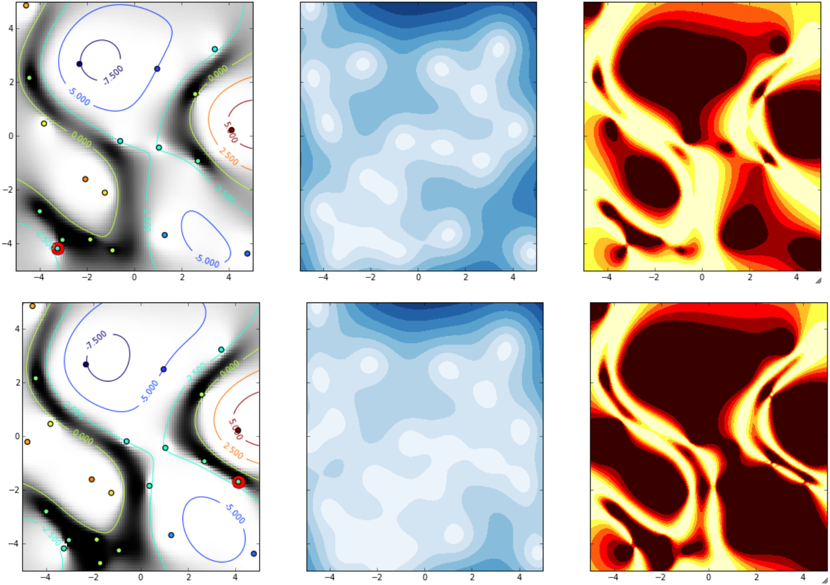


Iteration 15

Left$=p(\theta)$, middle$= E(\theta)$, right $= \tilde{J}(\theta)$



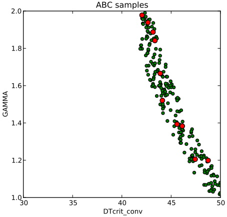
Iterations 20 and 24

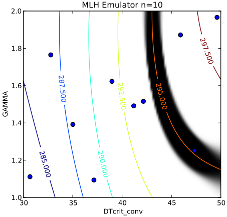


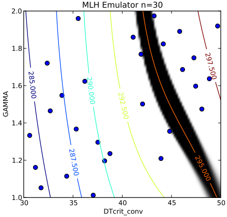
Video

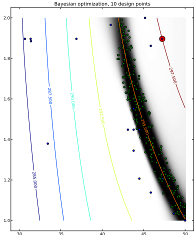
EPm: climate model

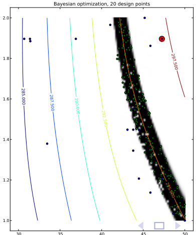
- 3d problem
- DTcrit_conv - critical temperature gradient that triggers convection
- GAMMA - emissivity parameter for water vapour
- Calibrate to global average surface temperature


ABC samples


MLH Emulator n=10


MLH Emulator n=30


Bayesian optimization, 10 design points


Bayesian optimization, 20 design points

Conclusions

- For complex models, surrogate-modelling approaches are often necessary
- Target of approximation: likelihood vs simulator output
  - likelihood is 1d surface, focussed on information in the data, but can be hard to model
  - Simulator output is multi-dimensional, and requires us to build a global approximation, and can be poorly modelled by a GP. But can be easier to model when Gaussian assumption appropriate.
- Good design can lead to substantial improvements in accuracy
  - Design needs to be specific to the task required - Space-filling designs are inefficient for calibration
  - Average entropy designs give good trade-off between exploration and defining the plausible region

Conclusions

- For complex models, surrogate-modelling approaches are often necessary
- Target of approximation: likelihood vs simulator output
  - likelihood is 1d surface, focussed on information in the data, but can be hard to model
  - Simulator output is multi-dimensional, and requires us to build a global approximation, and can be poorly modelled by a GP. But can be easier to model when Gaussian assumption appropriate.
- Good design can lead to substantial improvements in accuracy
  - Design needs to be specific to the task required - Space-filling designs are inefficient for calibration
  - Average entropy designs give good trade-off between exploration and defining the plausible region

Thank you for listening!