Design for Calibration and History Matching for Complex Simulators

Richard Wilkinson, James Hensman

School of Mathematics and Statistics, University of Sheffield
Medical School, University of Lancaster

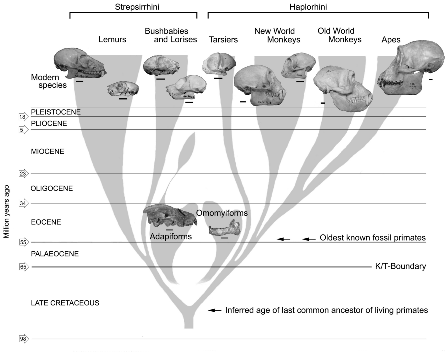
April 5, 2016

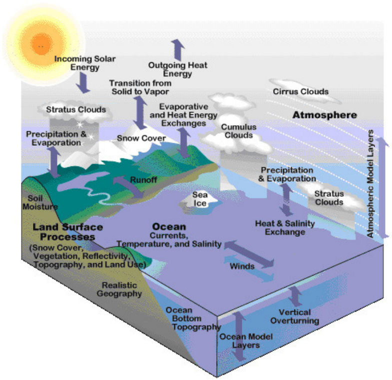
Outline

1. Calibration/history matching
2. ABC
3. Surrogate modelling
4. Design
   - Space filling designs are inefficient for calibration

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

$$\pi(\theta|D) \propto \pi(\theta)\pi(D|\theta)$$

for likelihood function
$\pi(D|\theta) = \int \pi(D|X,\theta)\pi(X|\theta)\mathrm{d}X$
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

$$\pi(\theta|D) \propto \pi(\theta)\pi(D|\theta)$$

for likelihood function

$\pi(D|\theta) = \int \pi(D|X,\theta)\pi(X|\theta)\mathrm{d}X$

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

$$\pi(\theta|D) \propto \pi(\theta)\pi(D|\theta)$$

for likelihood function
$\pi(D|\theta) = \int \pi(D|X,\theta)\pi(X|\theta)\mathrm{d}X$
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.

Approximate Bayesian Computation (ABC)

ABC algorithms are a collection of Monte Carlo methods used for calibrating stochastic 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 $\rho(D, X) \leq \epsilon$

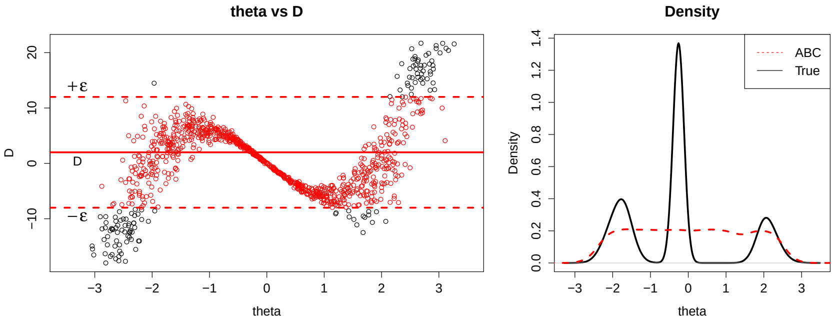
ϵ reflects the tension between computability and accuracy.

- 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 $\to$ Popular with non-statisticians

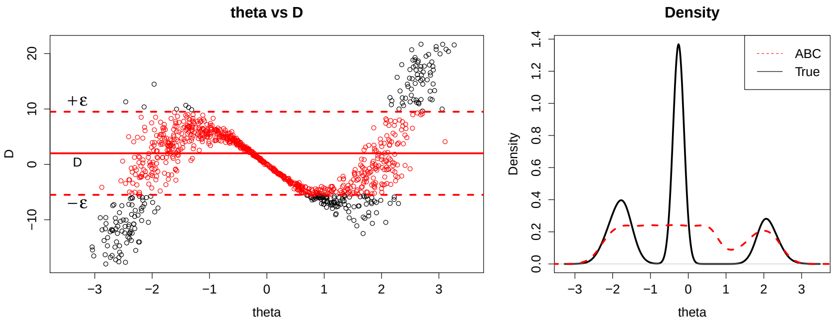
$\epsilon = 10$



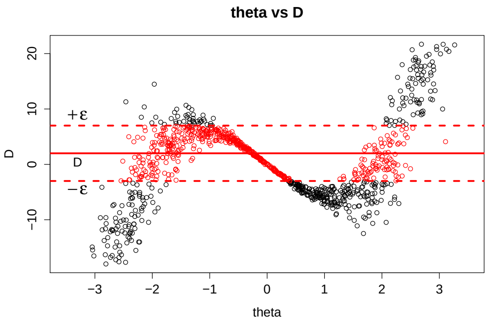
$$\theta \sim U[-10, 10], \qquad X \sim N(2(\theta+2)\theta(\theta-2), 0.1+\theta^2)$$

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

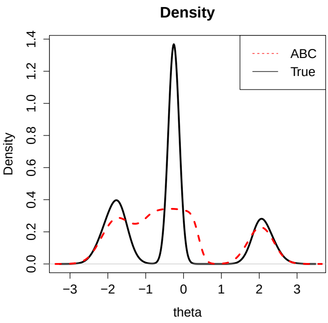
$\epsilon = 7.5$



$\epsilon = 5$

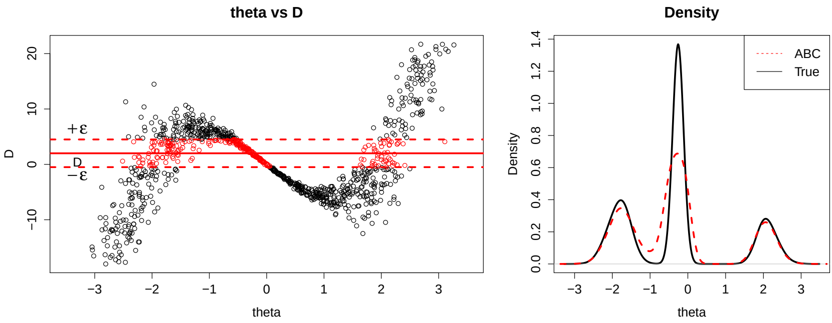


theta vs D
D
theta
+ε
D
−ε
20
10
0
−10
−3
−2
−1
0
1
2
3



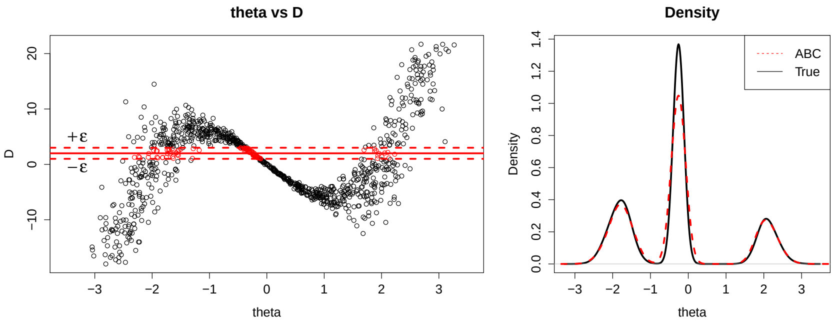
Density
Density
theta
ABC
True
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
−3
−2
−1
0
1
2
3

$\epsilon = 2.5$



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

$\epsilon = 1$


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

Surrogate modelling

If the model is expensive . . . use a surrogate model/emulator.

Surrogate modelling

If the model is expensive ... use a surrogate model/emulator.

What should we approximate with the surrogate model?

- simulator output

- Likelihood function

Surrogate modelling

If the model is expensive ... use a surrogate model/emulator.

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

Surrogate modelling

If the model is expensive . . . use a surrogate model/emulator.

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

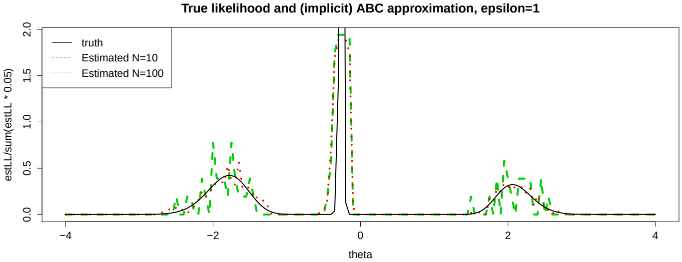
Wilkinson 2013

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

$$\pi_{ABC}(D|\theta) = \int \mathbb{I}_{\rho(D,X)<\epsilon}\pi(X|\theta)\mathrm{dX}$$

which we implicitly estimate using

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



Likelihood estimation

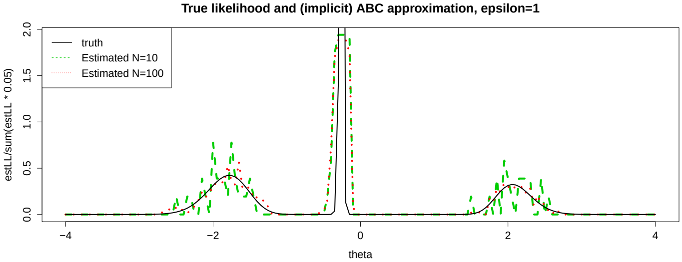
Wilkinson 2013

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

$$\pi_{ABC}(D|\theta) = \int \mathbb{I}_{\rho(D,X)<\epsilon}\pi(X|\theta)\mathrm{dX}$$

which we implicitly estimate using

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



Likelihood estimation

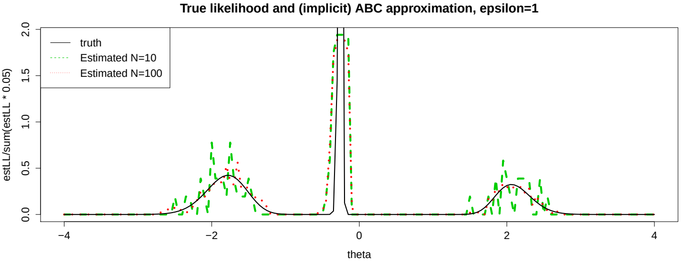
Wilkinson 2013

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

$$\pi_{ABC}(D|\theta) = \int \mathbb{I}_{\rho(D,X)<\epsilon}\pi(X|\theta)\mathrm{dX}$$

which we implicitly estimate using

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



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

Waves

We usually carry out history matching and ABC in a sequential manner

- Start with some larger than desired tolerance ϵ_0, find the plausible region
- Decrease the tolerance through a sequence of tolerances $\epsilon_0 \leq \epsilon_1 \leq \epsilon_n$ until the desired accuracy is achieved.

We are left needing to solve a sequence of classification problems.

Classification

In both history-matching and ABC, there is an element of classification, with parameters labelled as plausible or implausible, depending on the simulator output, ie, we try to find

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

where $\mathcal{P}$ is the set of plausible parameters.

Classification

In both history-matching and ABC, there is an element of classification, with parameters labelled as plausible or implausible, depending on the simulator output, ie, we try to find

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

where $\mathcal{P}$ is the set of plausible parameters.

- For history matching with deterministic simulators we often use something like
$$\mathcal{P} = \{\theta : \|f(\theta) - D\| \leq \delta\}$$

For probabilistic calibration, we can use a likelihood based criterion

$$\mathcal{P} = \{\theta : |l(\hat{\theta}) - l(\theta)| < T\}$$

where $l(\theta)$ is the log-likelihood, and $\hat{\theta}$ the mle. If we decide θ is implausible, we set

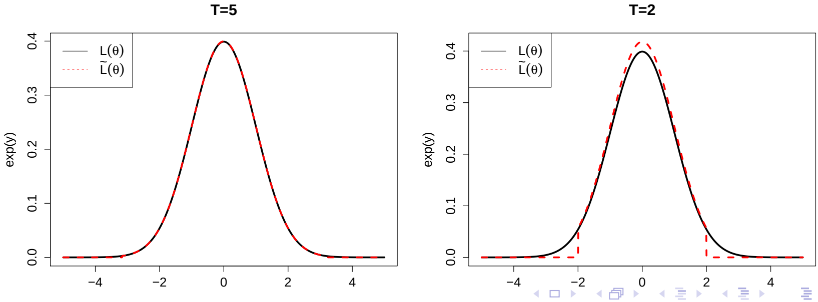
$$\pi(\theta|y) = 0$$

Using this criteria is equivalent to using the modified likelihood

$$\tilde{L}(\theta) \propto \exp(l(\theta))\mathbb{I}_{l(\hat{\theta})-l(\theta)<T}$$

Our hope is that

$$\tilde{\pi}(\theta|D) \approx \pi(\theta|D)$$



Design

The probability

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

is based upon a our GP model of the simulator or likelihood

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

Design

The probability

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

is based upon a our 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

- e.g., Maximin latin hypercubes, Sobol sequences

Design

The probability

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

is based upon a our 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

- e.g., 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 $\theta_1, \theta_2, \ldots$ 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

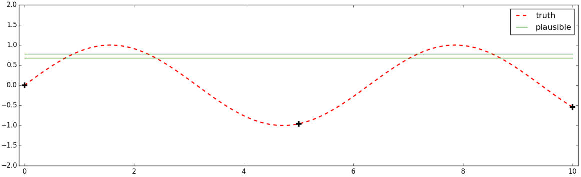
- The entropy of the classification surface is

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

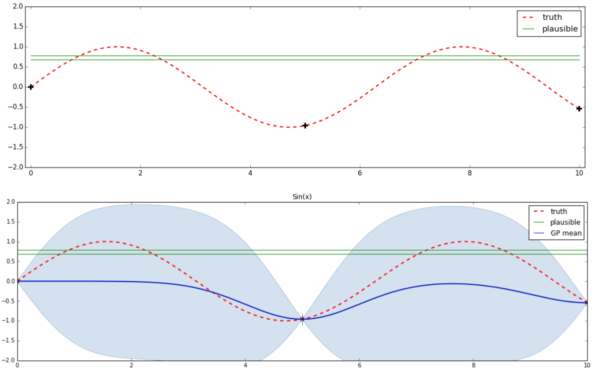
- Choose the next design point where we are most uncertain.

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

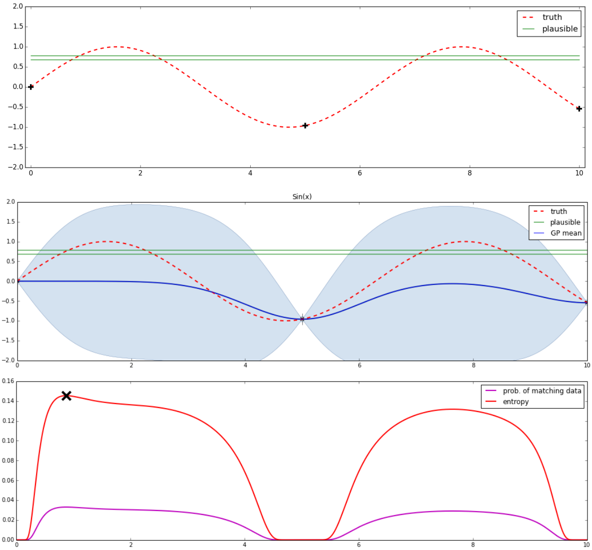
Toy 1d example $f(\theta) = \sin\theta$



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

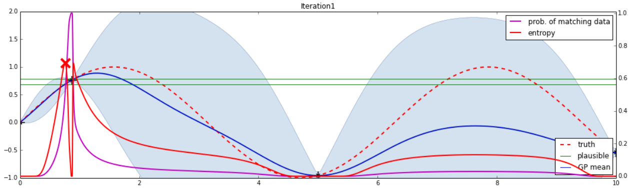


Toy 1d example $f(\theta) = \sin\theta$

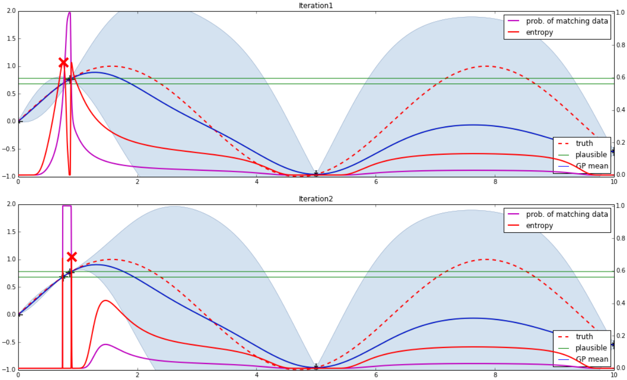


Add a new design point at the point of greatest entropy

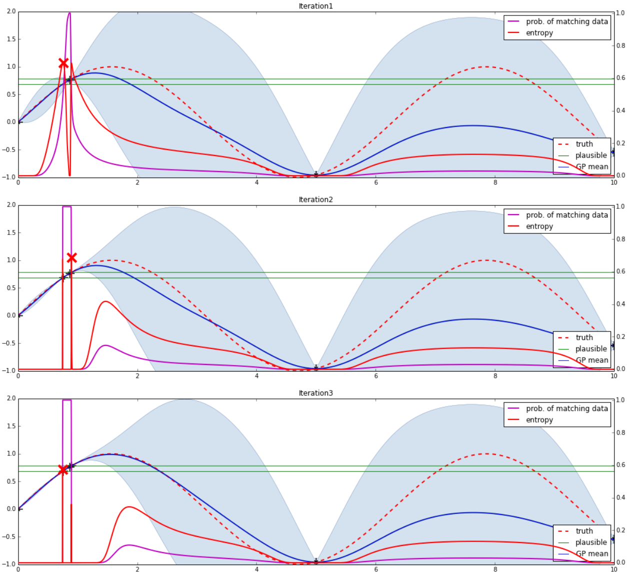
Toy 1d example $f(\theta) = \sin\theta$



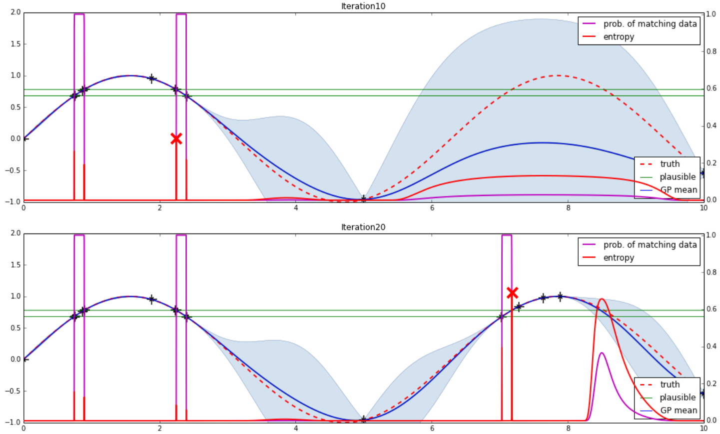
Toy 1d example $f(\theta) = \sin\theta$



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.

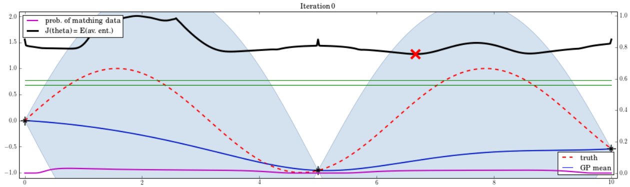
- Choose the next design point, θ_{n+1}, to minimise the expected average entropy

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

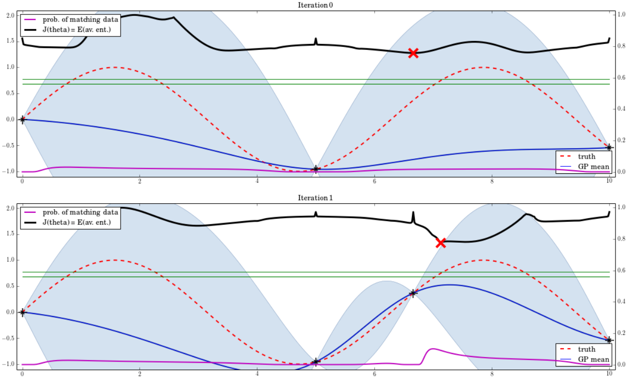
  where

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

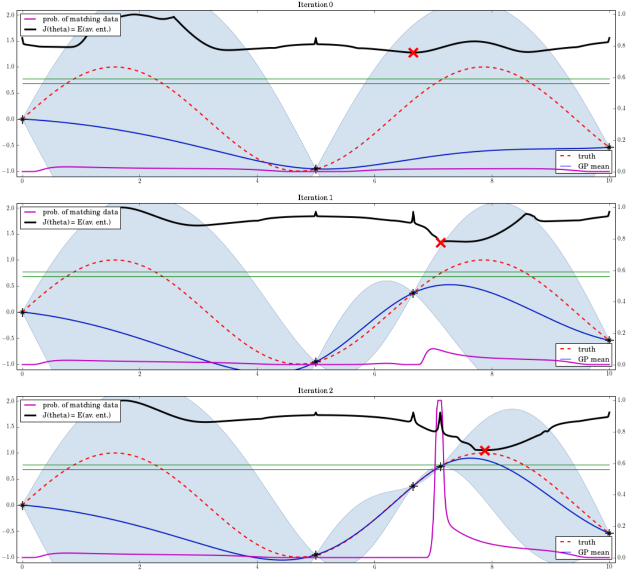
Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



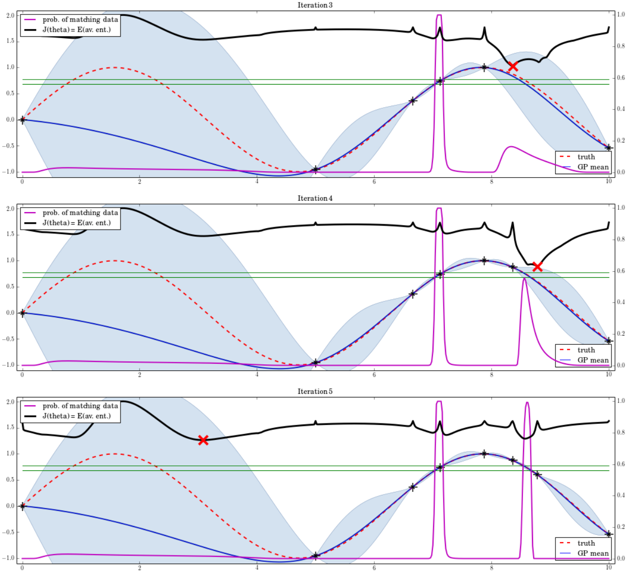
Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



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



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



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

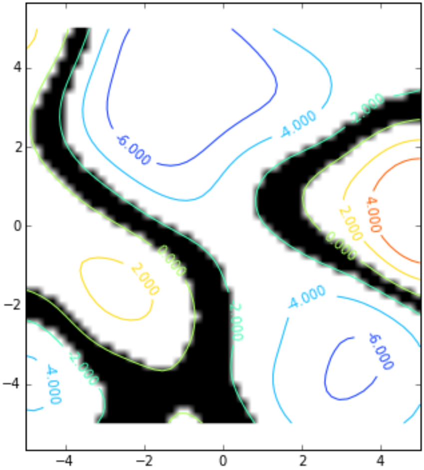
We can use Bayesian optimization to find the optima:

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

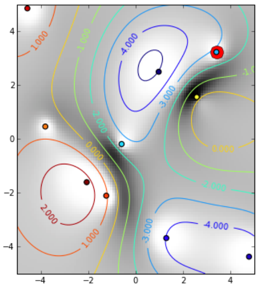
Can we learn the following plausible set?

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



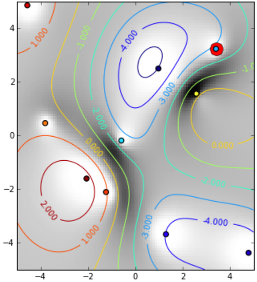
Iteration 10

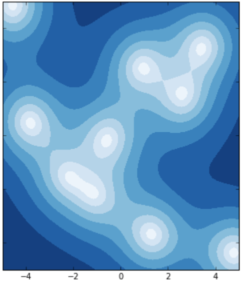
Left=$p(\theta)$, middle= $E(\theta)$, right = $\tilde{J}(\theta)$



Iteration 10

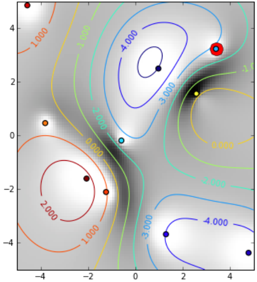
Left=$p(\theta)$, middle= $E(\theta)$, right = $\tilde{J}(\theta)$

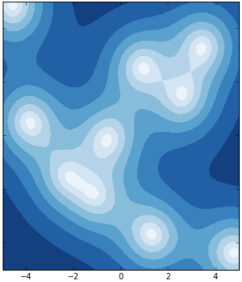


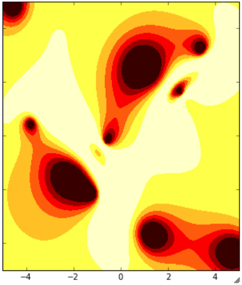


Iteration 10

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

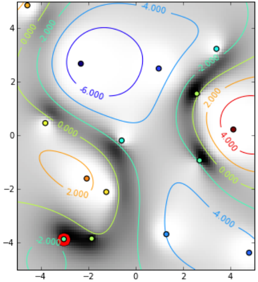


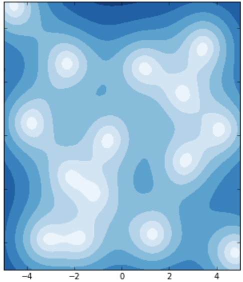


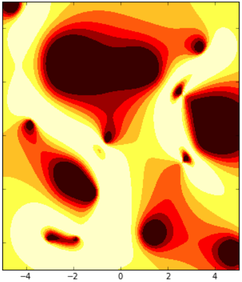


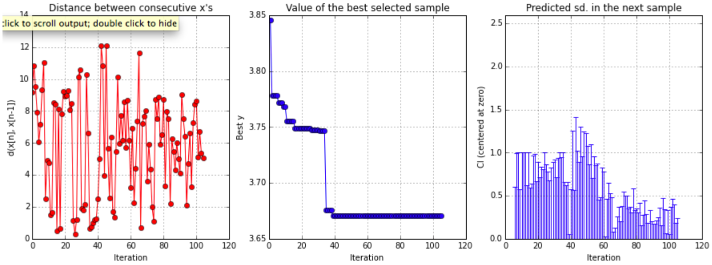
Iteration 15

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

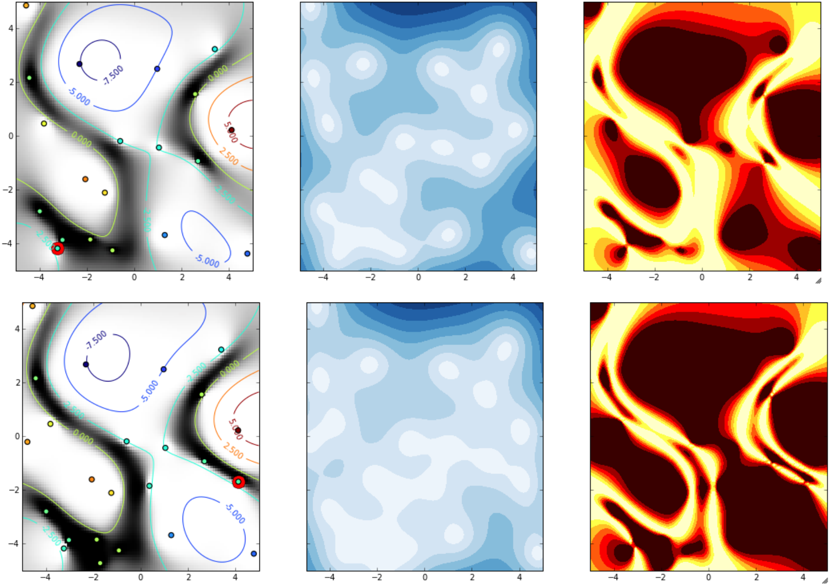








Iterations 20 and 24



Video

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!