Efficient history matching and calibration of complex simulators using Bayesian optimization

Richard Wilkinson, James Hensman

School of Mathematical Sciences
University of Nottingham/University of Sheffield

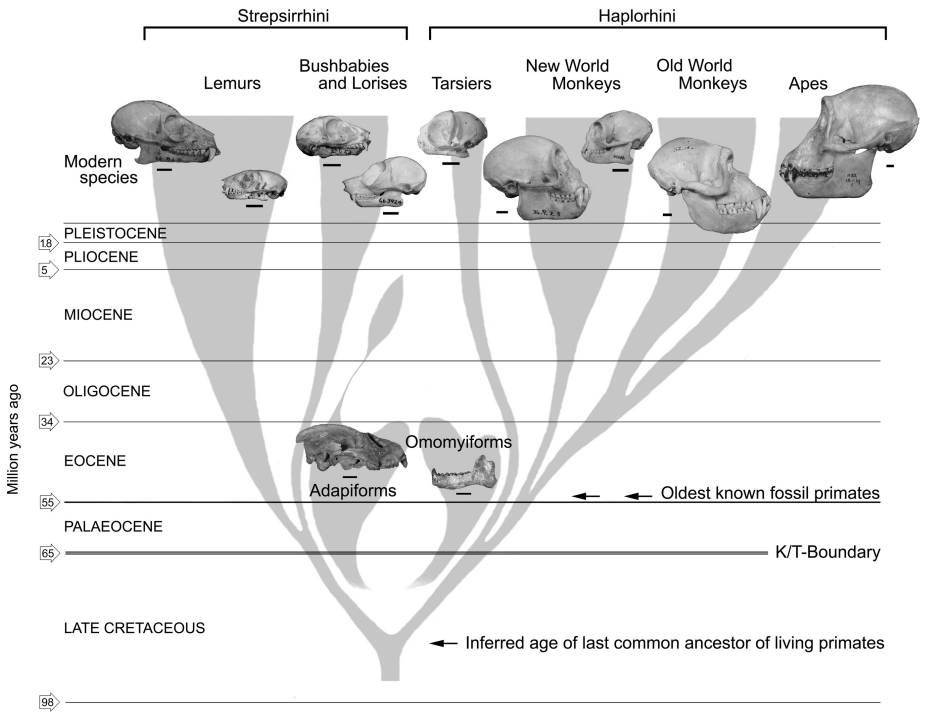
August 10, 2015

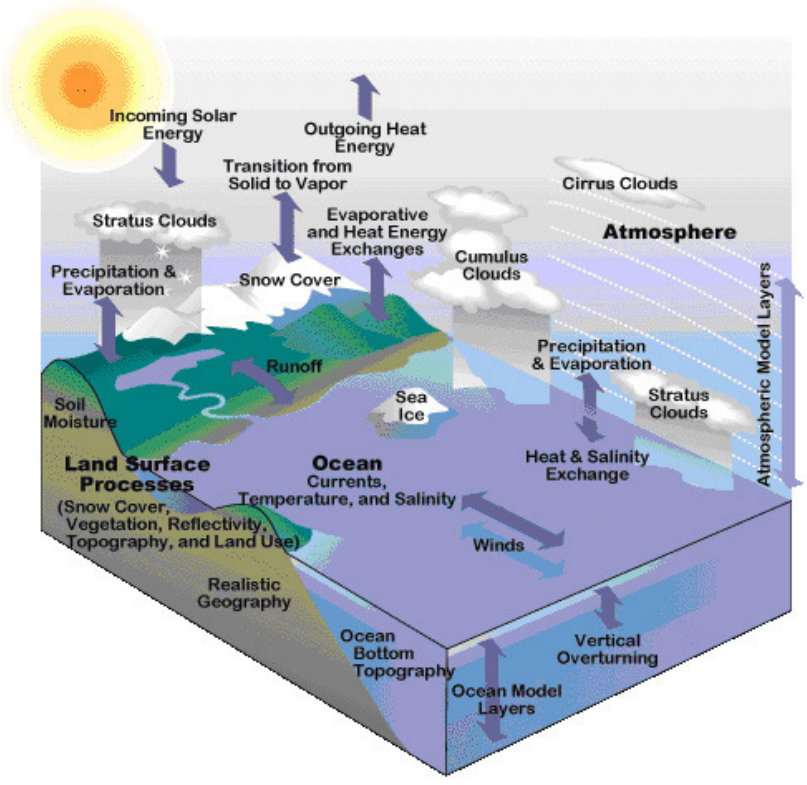
Outline

1. Calibration/history matching
2. ABC and emulation
3. Entropic designs

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

History matching

Find the plausible parameter set $\mathcal{P}_\theta$:

$$f(\theta) \in \mathcal{P}_D \; \forall \; \theta \in \mathcal{P}_\theta$$

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 aims to find a distribution representing plausible parameter values, whereas history matching classifies space as plausible or implausible.

Intractability

For complex problems we often run into difficulties:

- Computational intractability/code uncertainty: if $f(\theta)$ is expensive to evaluate, then we can only evaluate $\pi(D|\theta)$ at a small number of θ values.
  - e.g., climate models, ground-water flow problems, engineering ...
- Mathematical intractability: for many complex stochastic simulators $\pi(D|\theta)$ can't be evaluated (unknown is subjective). I.e., if the analytic distribution of the simulator, $f(\theta)$, run at θ is unknown.
  - Typical in biological problems, particularly genetics, epidemiology, ecology etc.

Emulators for deterministic simulators

Sacks *et al.* 1989, Kennedy and O'Hagan 2001

- If $f(\theta)$ is expensive to evaluate, then we can only afford a limited ensemble of simulator evaluations

$$D = \{\theta_i, f(\theta_i)\}_{i=1}^{n}$$

- We are uncertain about $f(\theta)$ for θ not in the design

Emulators for deterministic simulators

Sacks *et al.* 1989, Kennedy and O'Hagan 2001

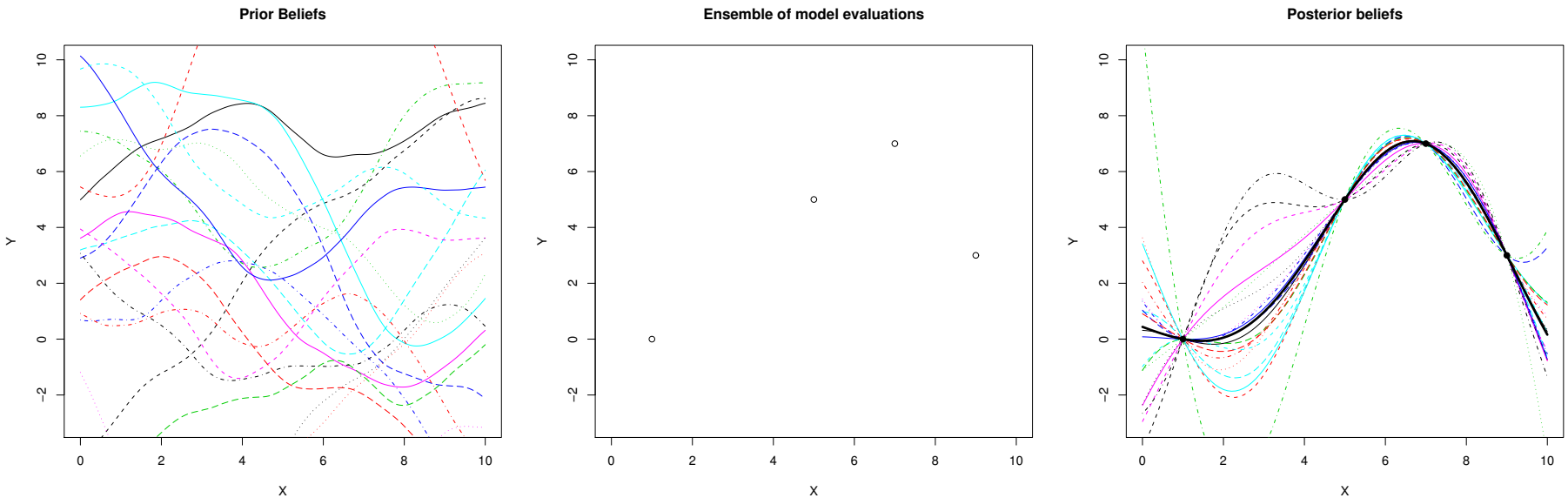
- If $f(\theta)$ is expensive to evaluate, then we can only afford a limited ensemble of simulator evaluations

$$D = \{\theta_i, f(\theta_i)\}_{i=1}^{n}$$

- We are uncertain about $f(\theta)$ for θ not in the design

An emulator is a cheap statistical surrogate $\tilde{f}(\theta)$ which approximates $f(\theta)$.

Gaussian processes (GP) are a common choice: $\tilde{f}(\cdot) \sim GP(m(\cdot), c(\cdot,\cdot))$



We can then use $\tilde{f}$ in place of f in any analysis.

History-matching

Craig *et al.* 2001, Vernon *et al.* 2010

History matching is used in the analysis of computer experiments to rule out regions of space as implausible.

1. Build an emulator using a design of simulator runs set
2. Relate the simulator to the data where ϵ

$$D = \zeta + e + \epsilon$$

where ϵ is simulator discrepancy and e is measurement error

3. Declare θ implausible if, e.g.,

$$\| D - \mathbb{E}\tilde{f}(\theta) \| > 3\sigma$$

where σ^2 is the combined variance implied by the emulator, discrepancy and measurement error.

History-matching

Craig *et al.* 2001, Vernon *et al.* 2010

History matching is used in the analysis of computer experiments to rule out regions of space as implausible.

1. Build an emulator using a design of simulator runs set
2. Relate the simulator to the data where ϵ

$$D = \zeta + e + \epsilon$$

   where ϵ is simulator discrepancy and e is measurement error
3. Declare θ implausible if, e.g.,

$$\| D - \mathbb{E}\tilde{f}(\theta) \| > 3\sigma$$

   where σ^2 is the combined variance implied by the emulator, discrepancy and measurement error.
4. Add more design points in plausible region (more simulator runs), build a better emulator
5. $\vdots$

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

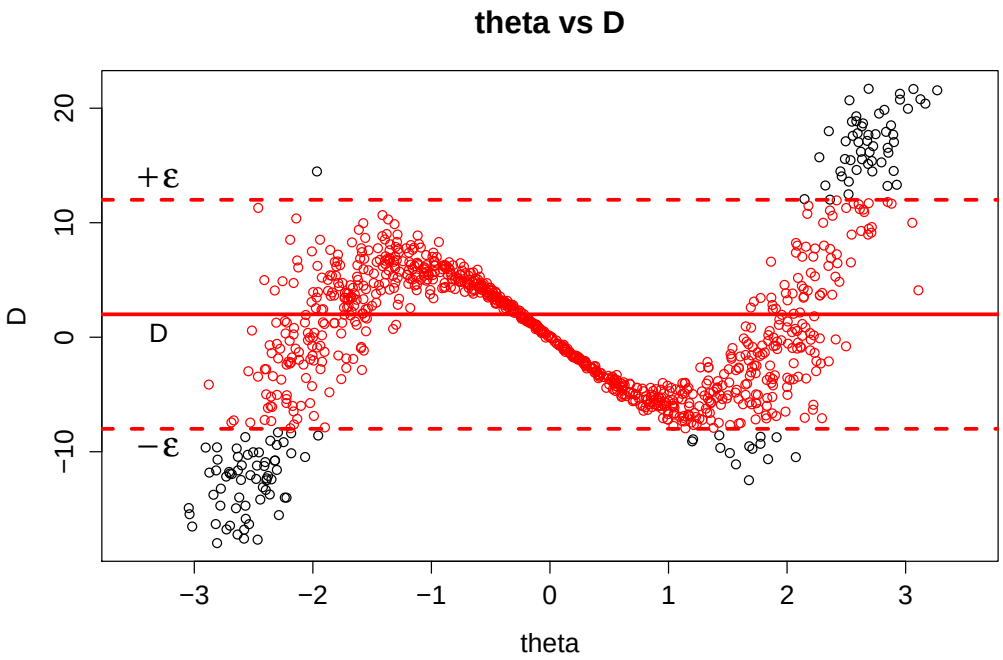
- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(D, X) \leq \epsilon$

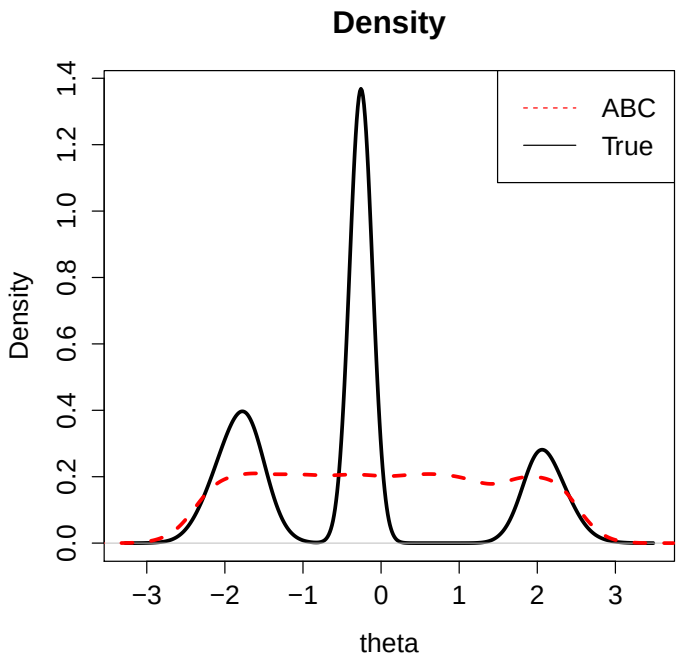
ϵ reflects the tension between computability and accuracy.

- 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.

$\epsilon = 10$

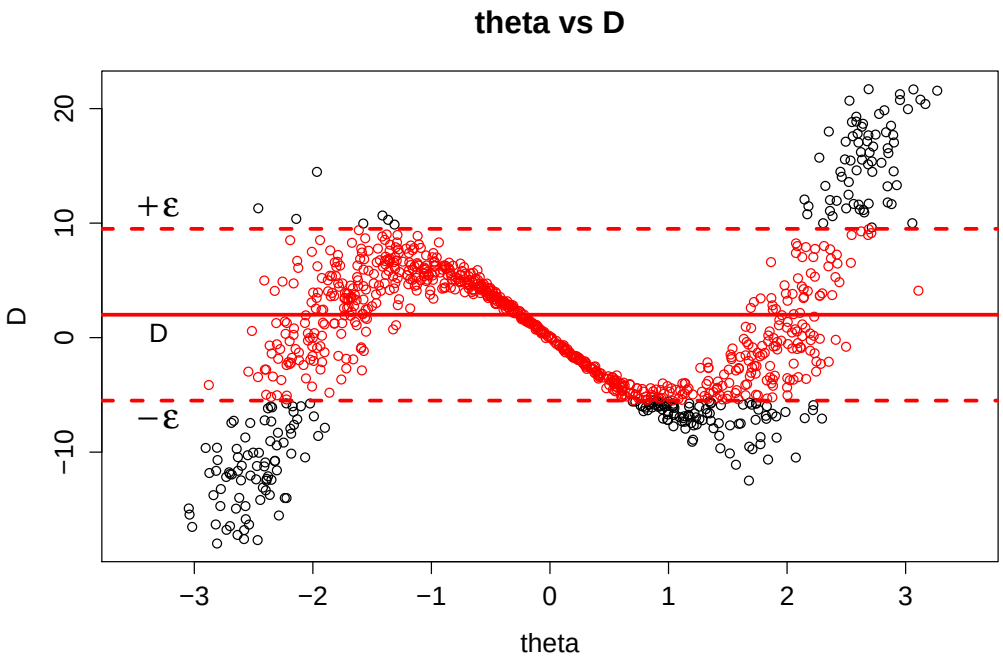


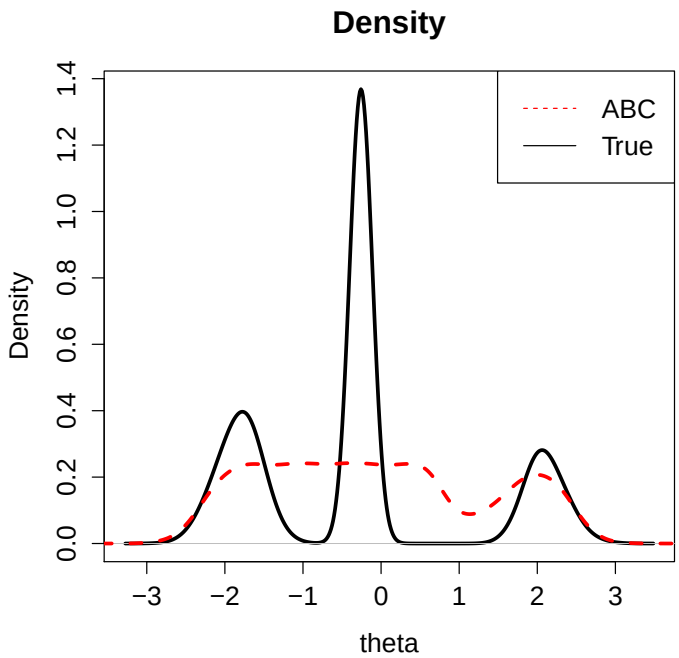


$$\theta \sim U[-10, 10], \qquad X \sim N(2(\theta + 2)\theta(\theta - 2), 0.1 + \theta^2)$$

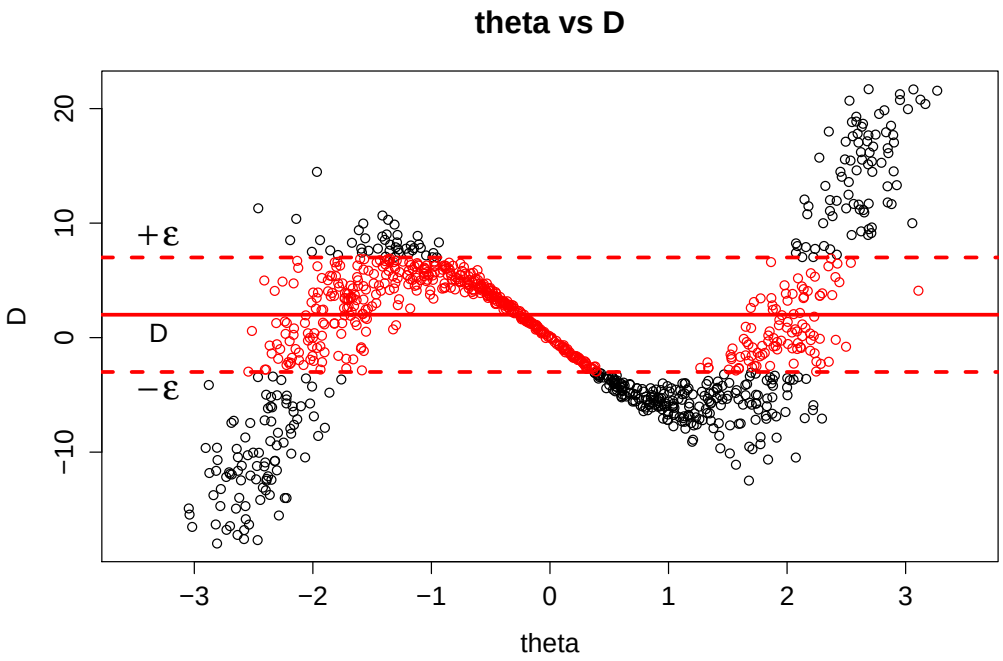
$$\rho(D, X) = |D - X|, \qquad D = 2$$

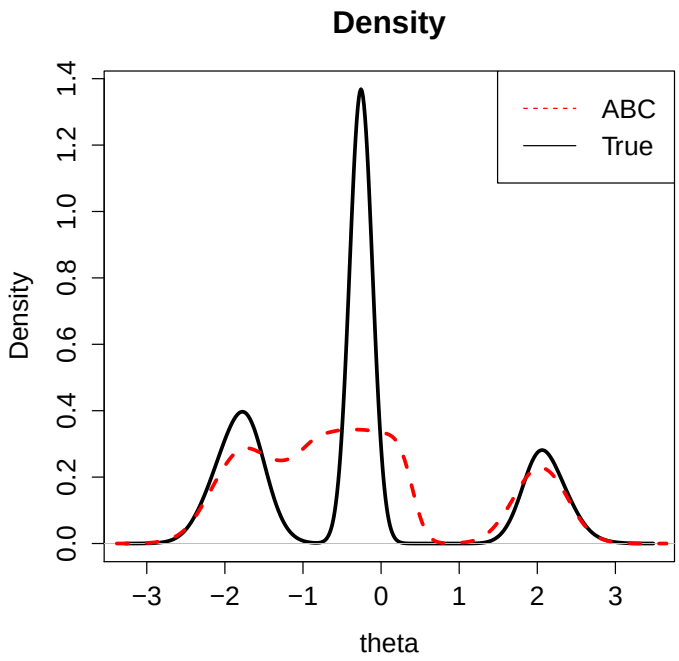
$\epsilon = 7.5$


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

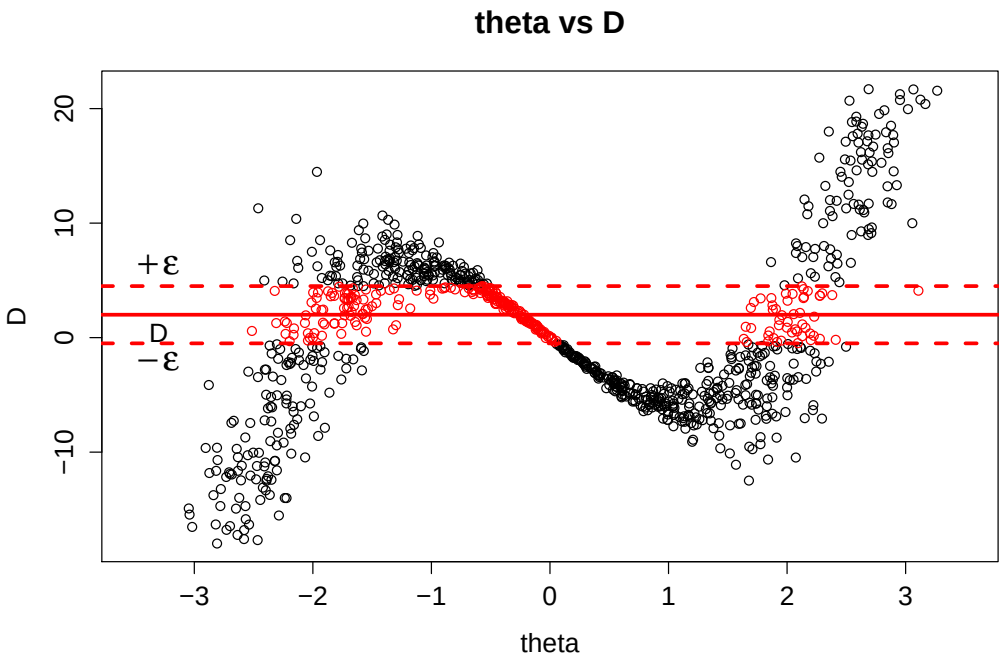

Density
Density
theta
ABC
True
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3

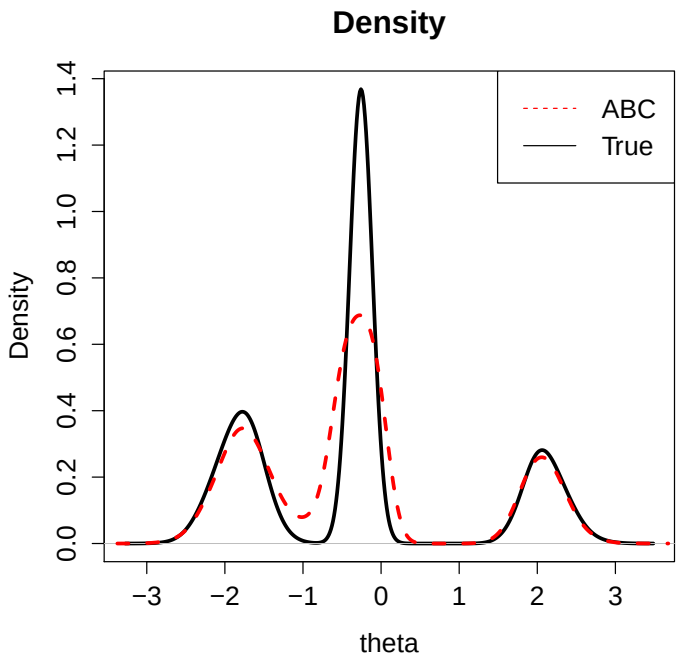
$\epsilon = 5$


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

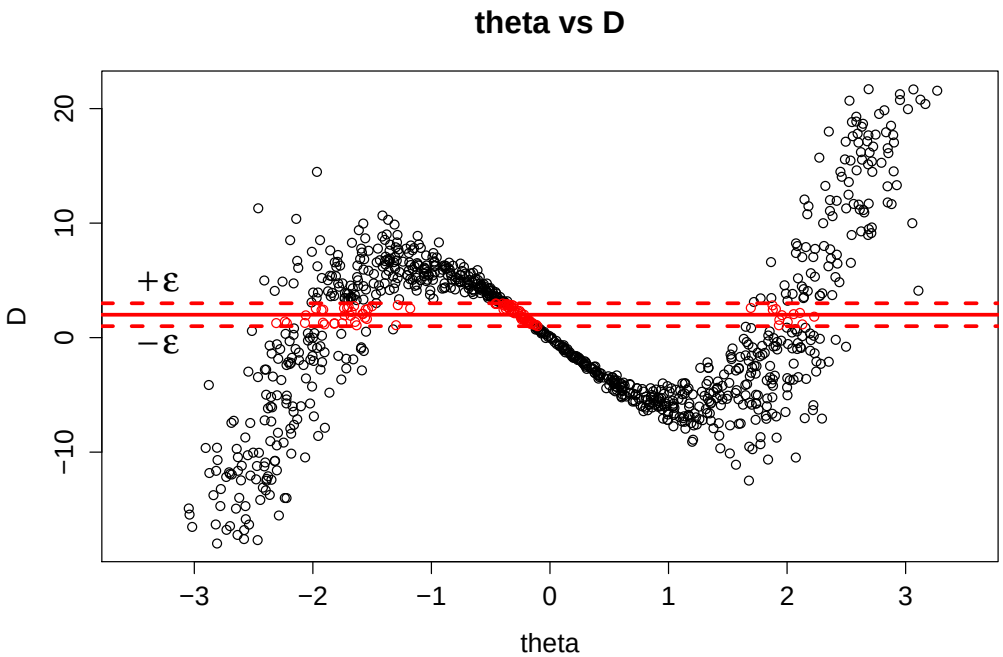

Density
Density
theta
ABC
True
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3

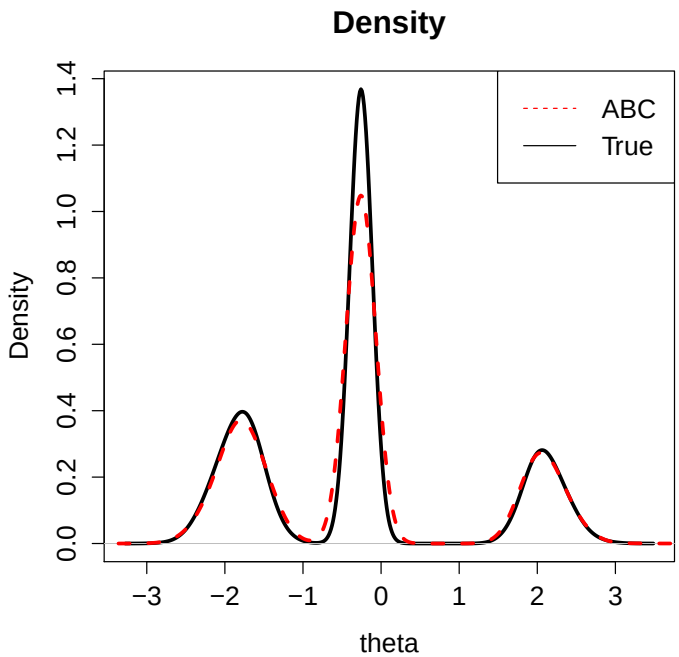
$\epsilon = 2.5$


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


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

$\epsilon = 1$


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


Density
Density
theta
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

Limitations of Monte Carlo methods

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

This guarantee is costly and can require more simulation than is possible.

Limitations of Monte Carlo methods

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

This guarantee is costly and can require more simulation than is possible.

However,

- 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 careful design.

We can use methods that don't suffer in this way, but at the cost of losing the guarantee of success.

Likelihood estimation

Wilkinson 2013

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

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

where $\pi(D|X)$ is the ABC acceptance kernel.

Likelihood estimation

Wilkinson 2013

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

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

where $\pi(D|X)$ is the ABC acceptance kernel.

We can estimate this using repeated runs from the simulator

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

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

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

where $\pi(D|X)$ is the ABC acceptance kernel.

We can estimate this using repeated runs from the simulator

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

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

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

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

History matching waves

Wilkinson 2014

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

$$l(\theta) = \log L(\theta)$$

History matching waves

Wilkinson 2014

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

$$l(\theta) = \log L(\theta)$$

However, the log-likelihood for a typical problem ranges across too wide a range of values.

Consequently, most GP models will struggle to model the log-likelihood across the parameter space.

History matching waves

Wilkinson 2014

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

$$l(\theta) = \log L(\theta)$$

However, the log-likelihood for a typical problem ranges across too wide a range of values.

Consequently, most GP models will struggle to model the log-likelihood across the 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 likelihood is too difficult to model, so we model the log-likelihood instead.

$$l(\theta) = \log L(\theta)$$

However, the log-likelihood for a typical problem ranges across too wide a range of values.

Consequently, most GP models will struggle to model the log-likelihood across the parameter space.

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

When this works, it can give huge savings in the number of simulator runs required.

Implausibility

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

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

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

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

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

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

Usual design choices are space filling designs

- Maximin latin hypercubes, Sobol sequences

Entropic designs

However, space filling designs are good for global approximations, but wasteful for history-matching.

- 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

However, space filling designs are good for global approximations, but wasteful for history-matching.

- 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

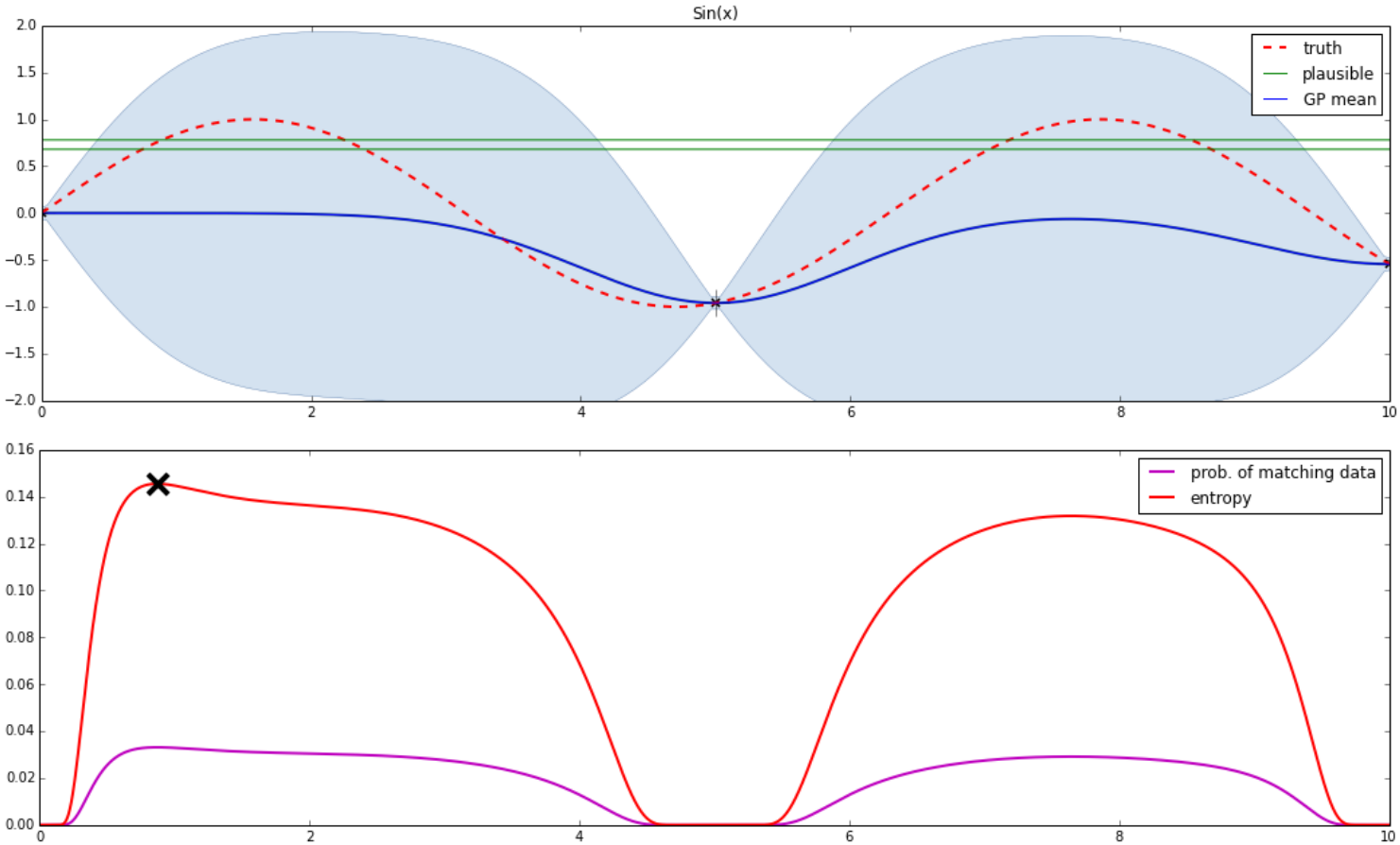
- The entropy of the classification surface is

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

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

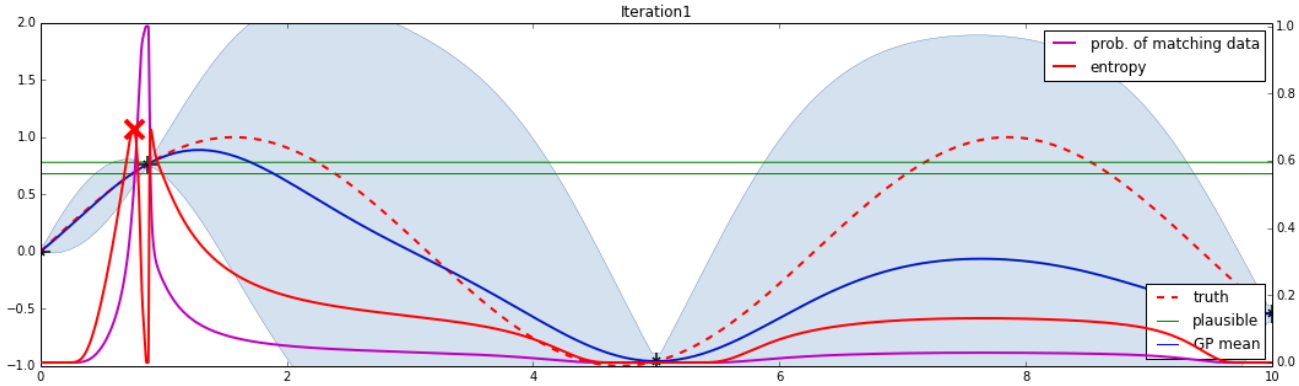
$$\theta_{n+1} = \arg\max E(\theta)$$

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

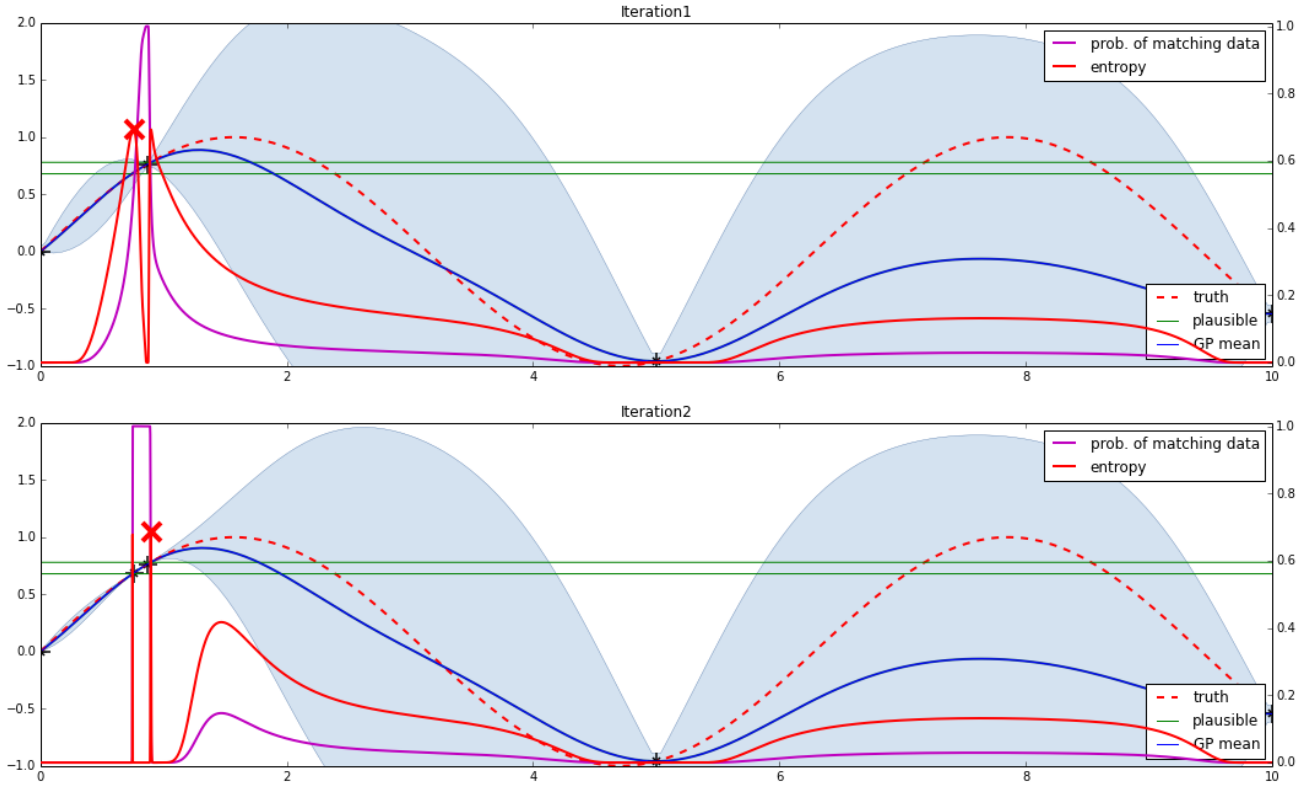


Add a new design point (simulator evaluation) at the point of greatest entropy

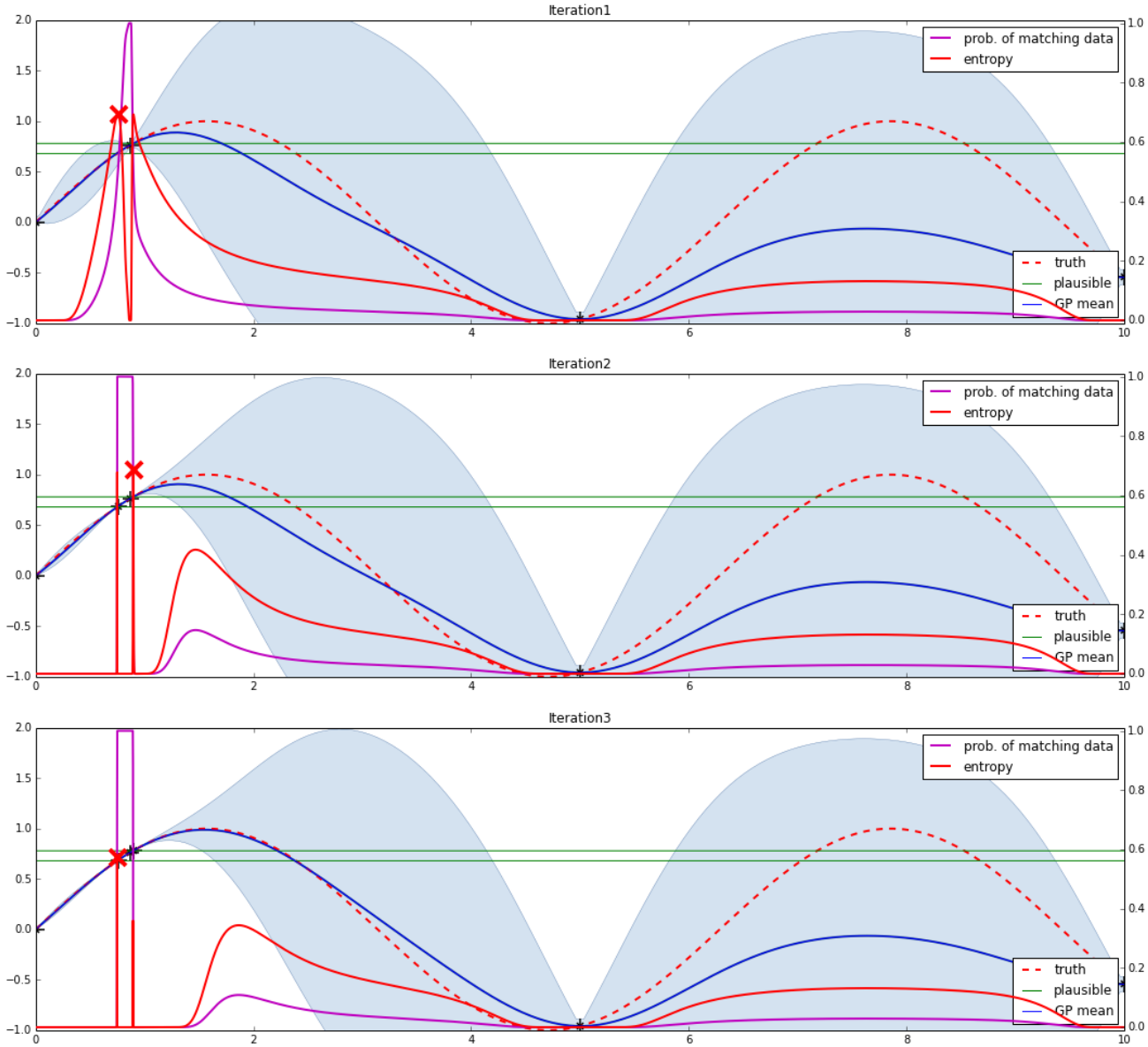
Toy 1d example $f(\theta) = \sin\theta$



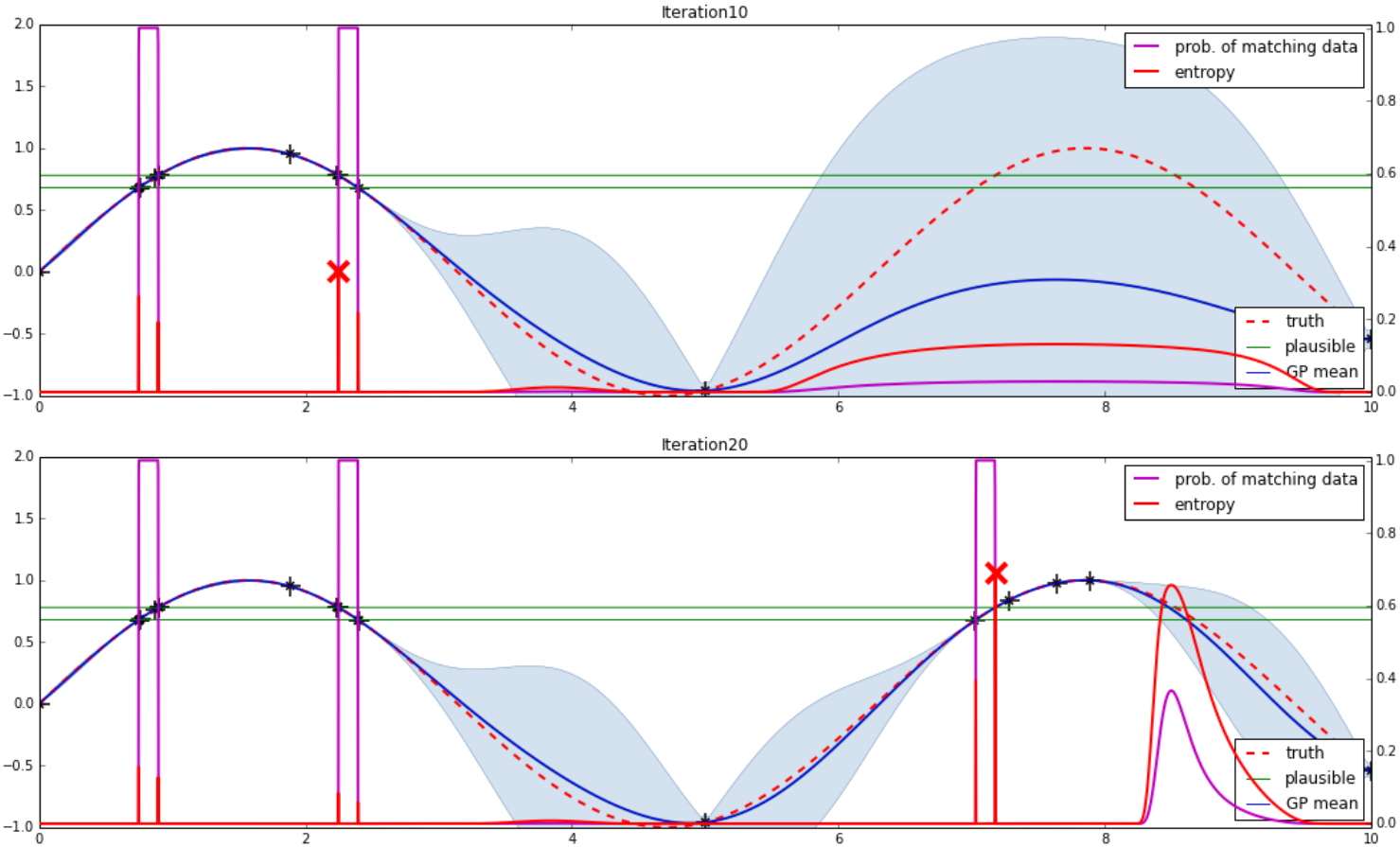
Toy 1d example $f(\theta) = \sin\theta$



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.

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.

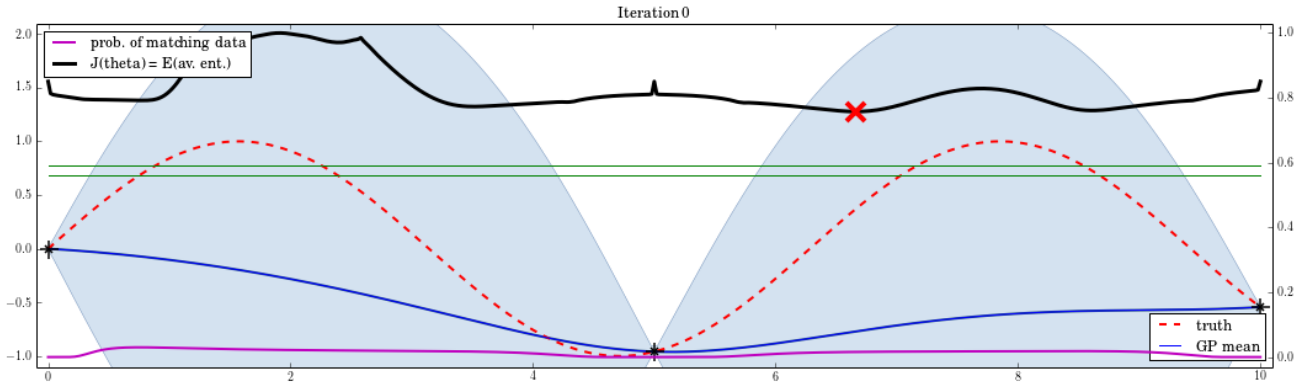
- Choose the next design point, θ_{n+1}, to minimise the expected average entropy

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

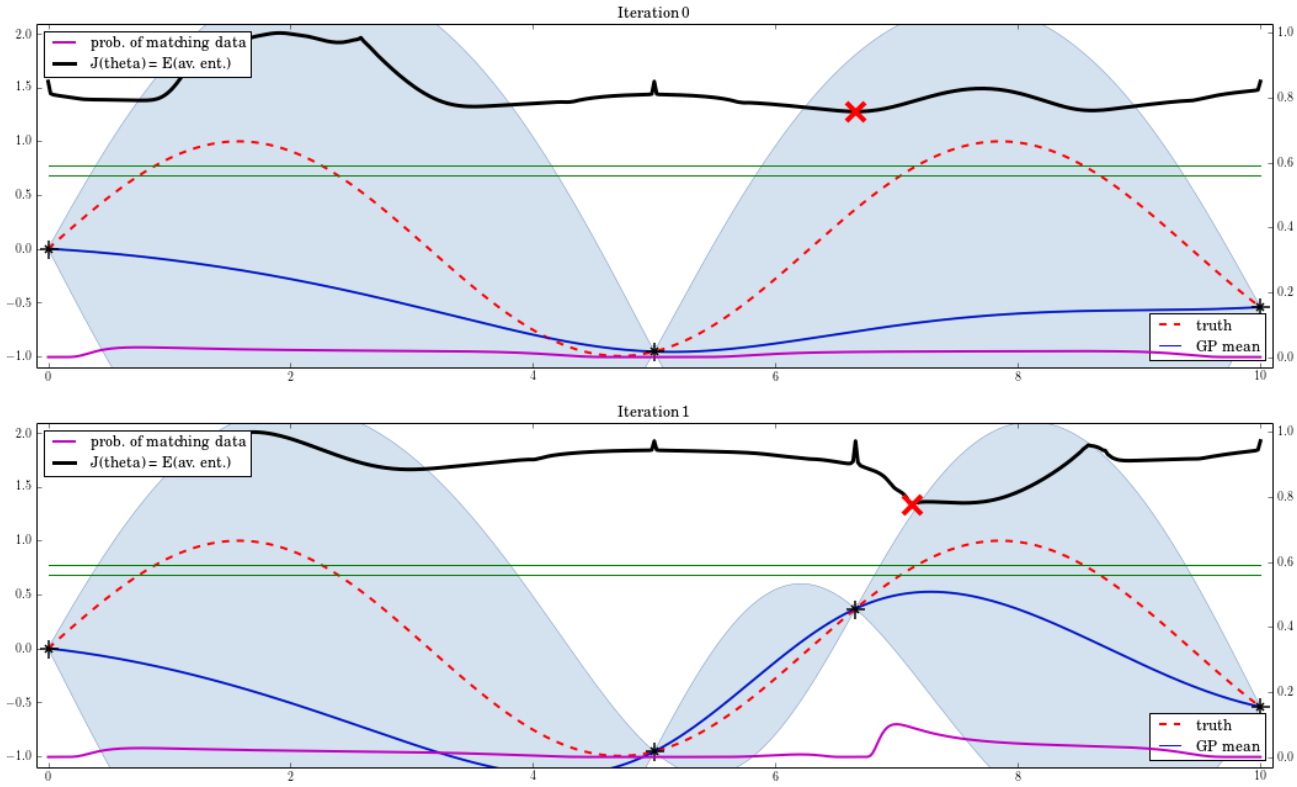
  where

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

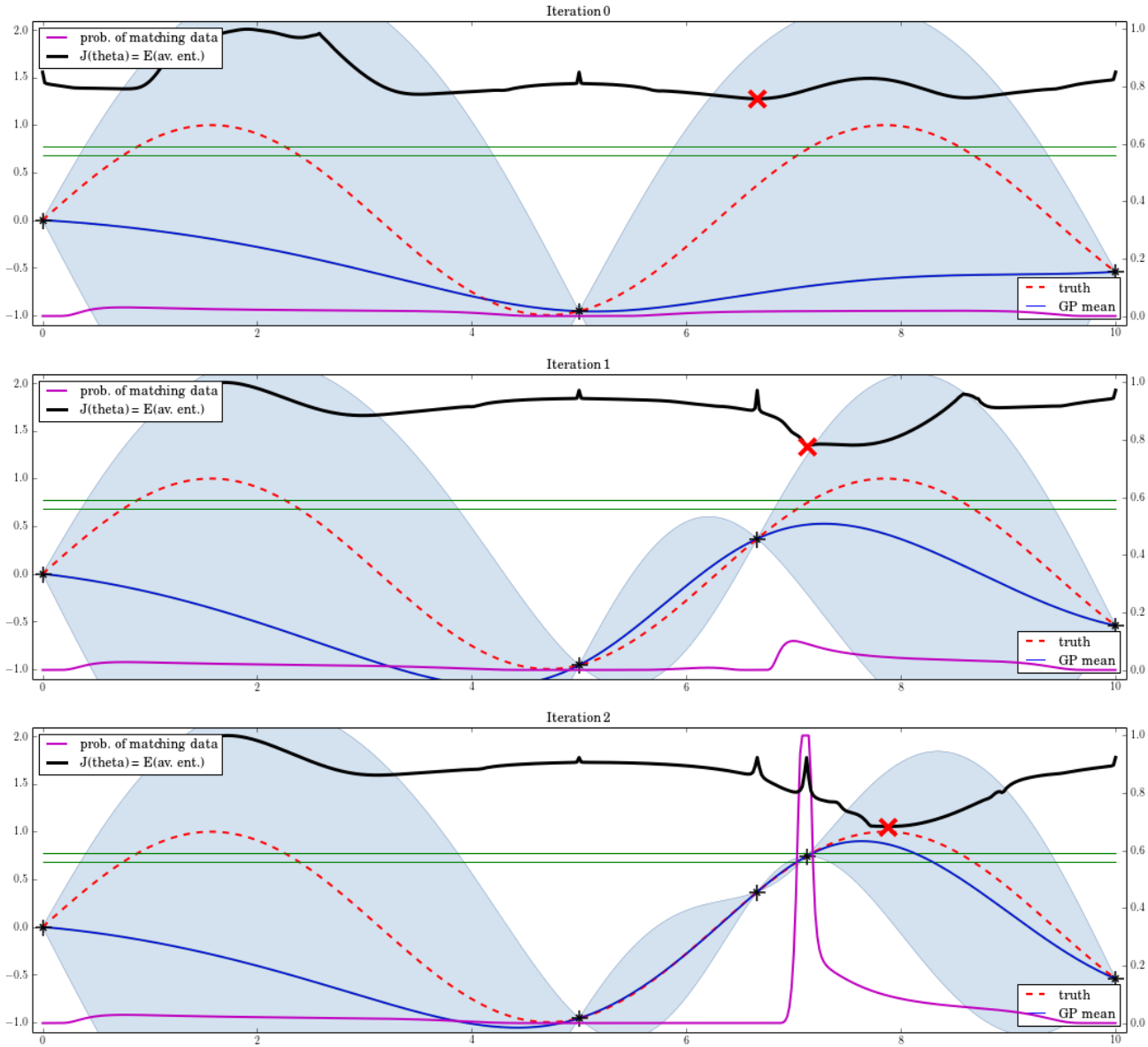
Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



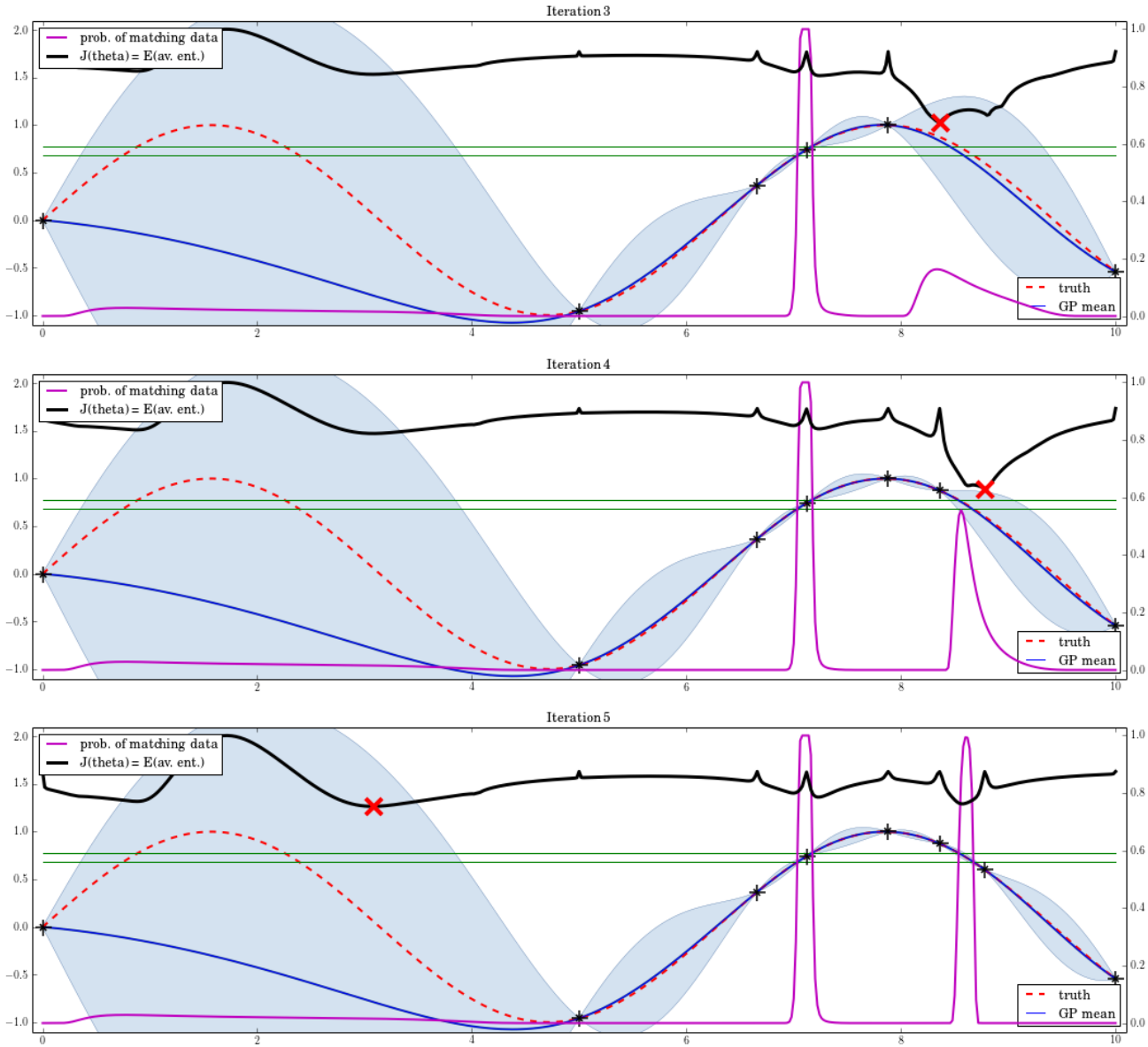
Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



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

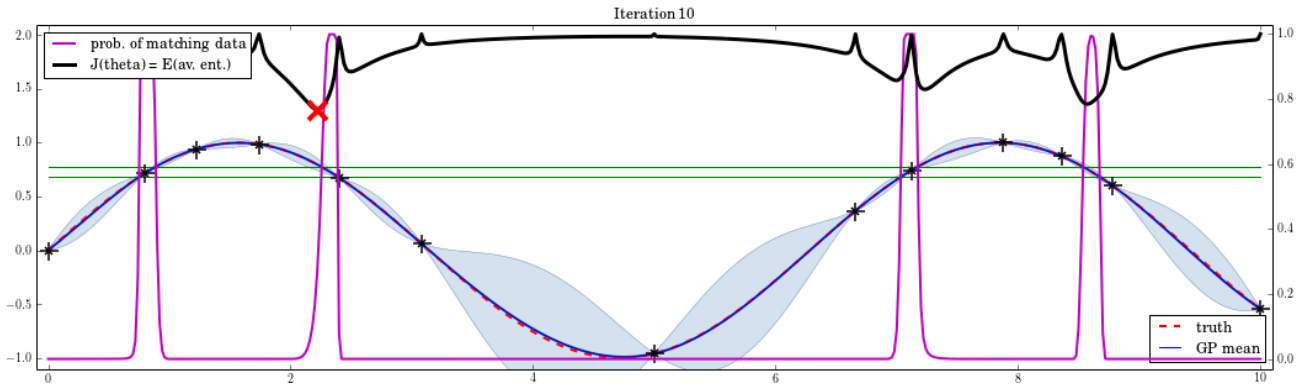


Toy 1d example $f(\theta) = \sin\theta$ - Expected entropy



Toy 1d: min expected entropy vs max entropy

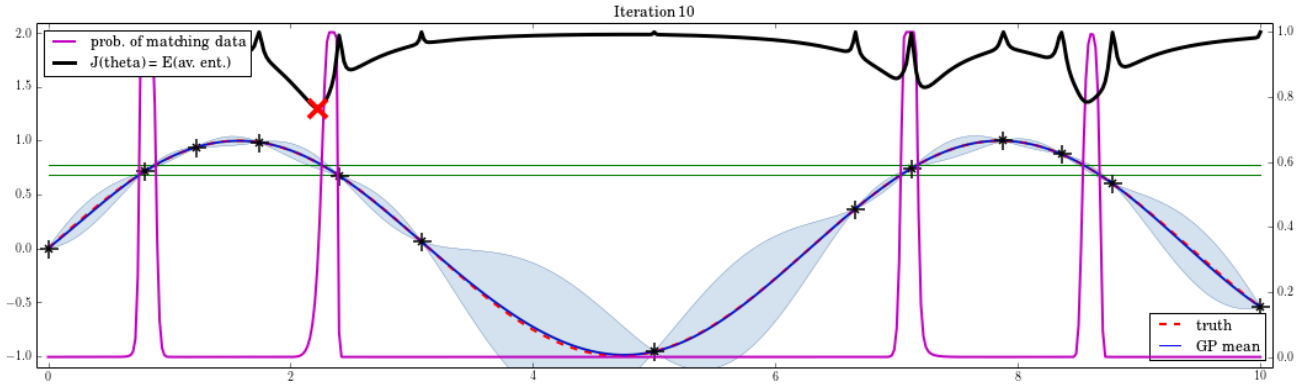
After 10 iterations, choosing the point of maximum entropy



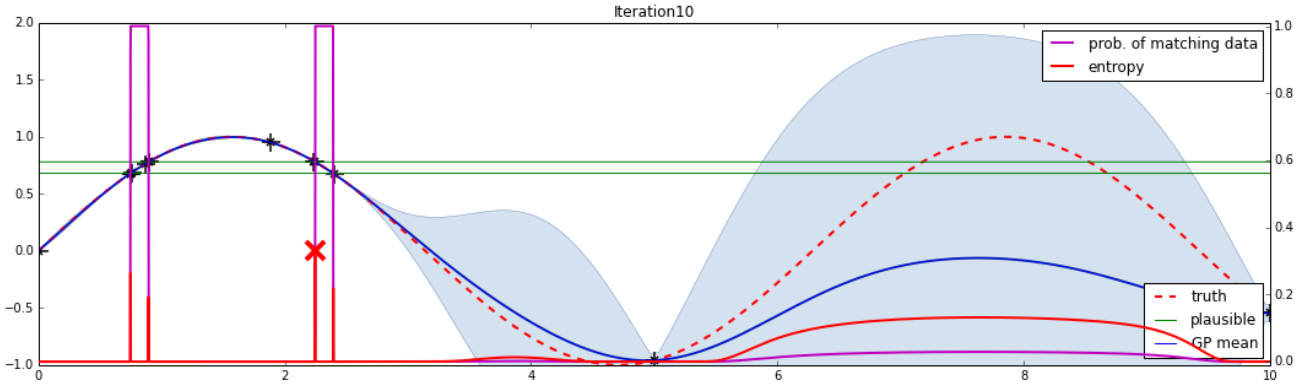
we have found the plausible region to reasonable accuracy.

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

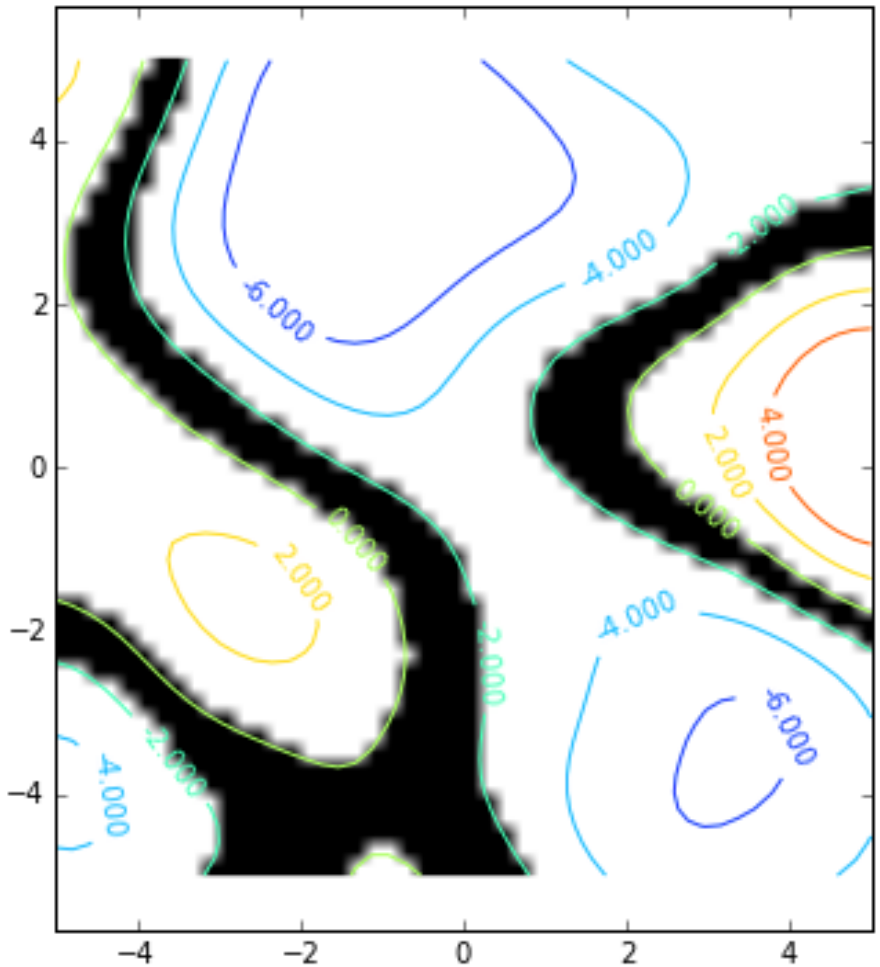
We can use Bayesian optimization to find the optima:

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

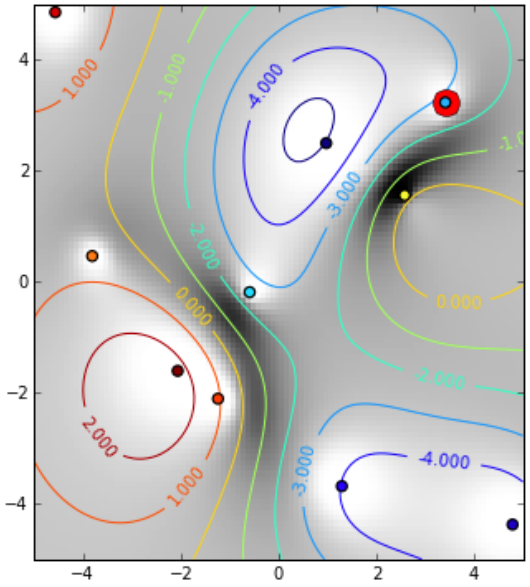
Can we learn the following plausible set?

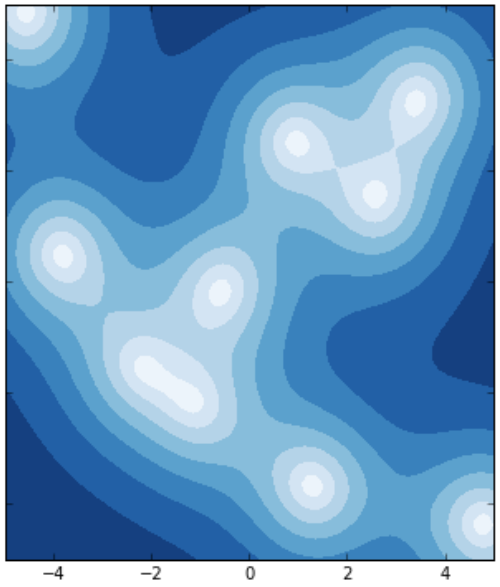
- A sample from a GP on $\mathbb{R}^2$.
- Find x s.t. $-2 < f(x) < 0$

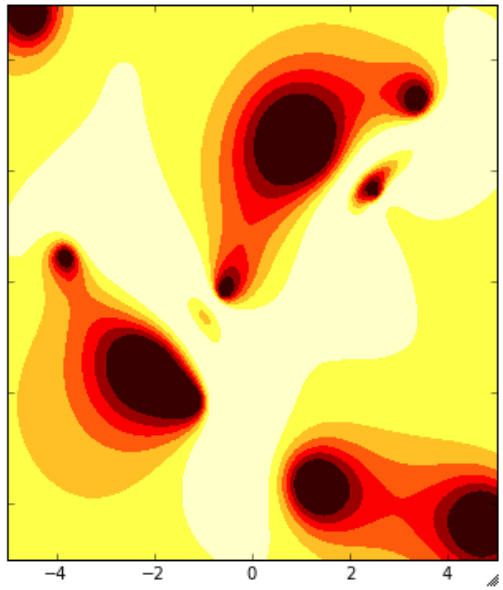


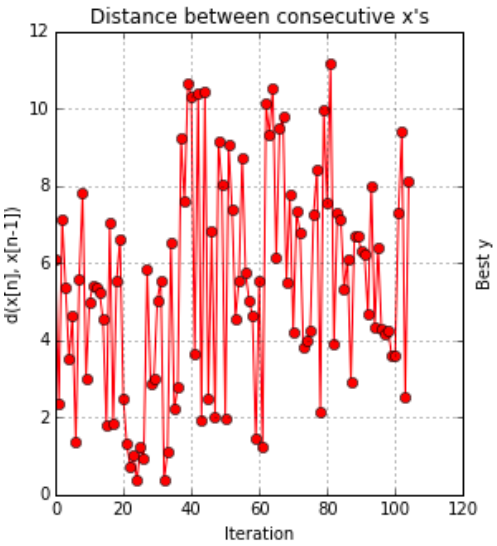
Iteration 10

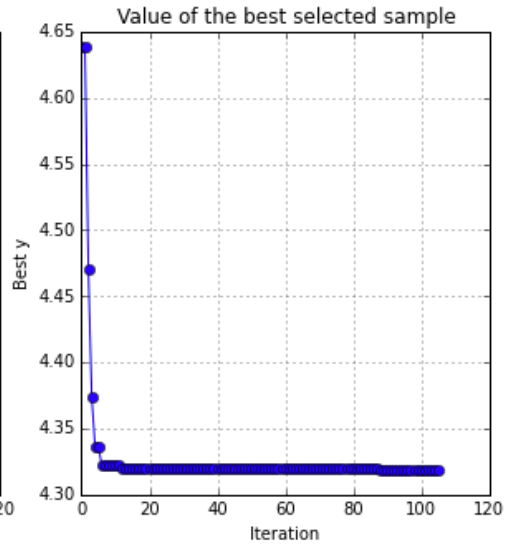
Left=$p(\theta)$, middle= $E(\theta)$, right = $\tilde{J}(\theta)$

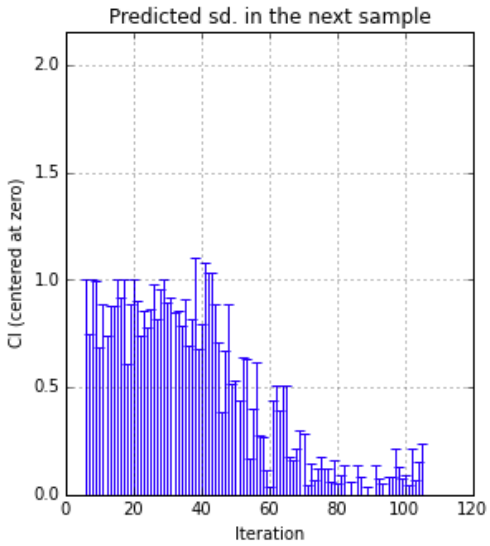






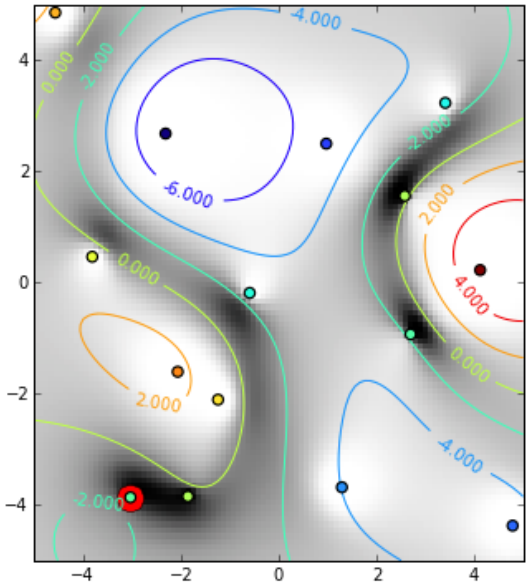


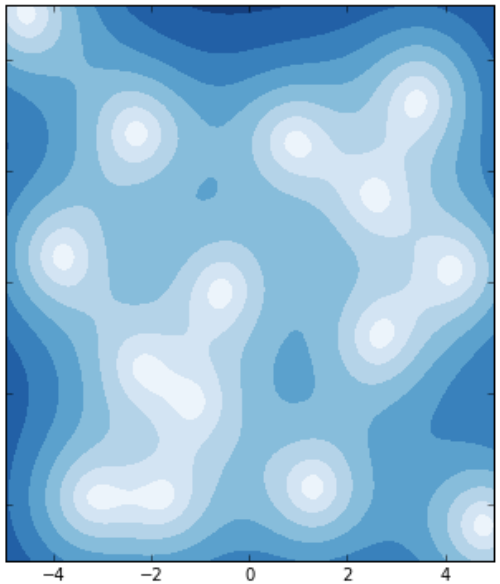


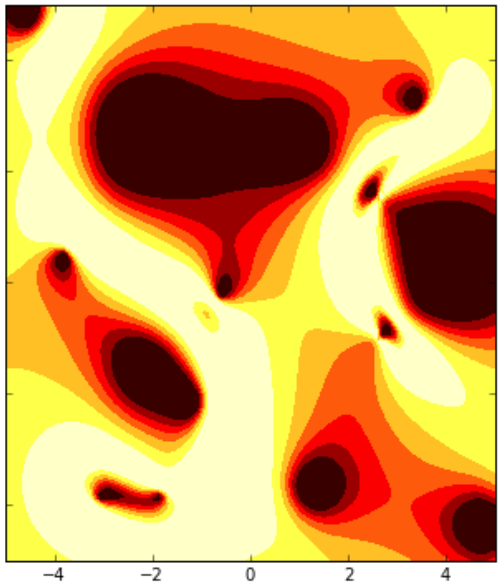


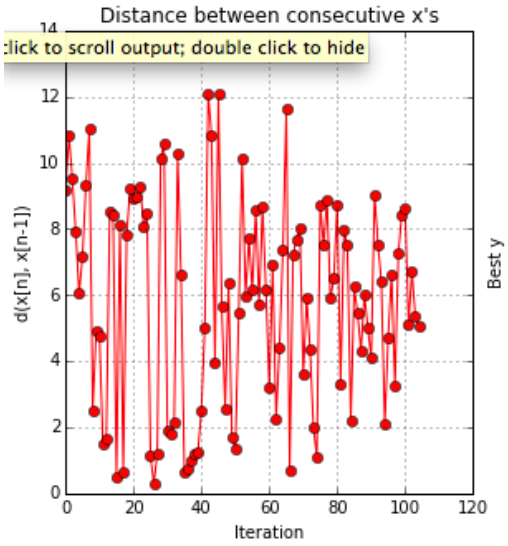
Iterations 15

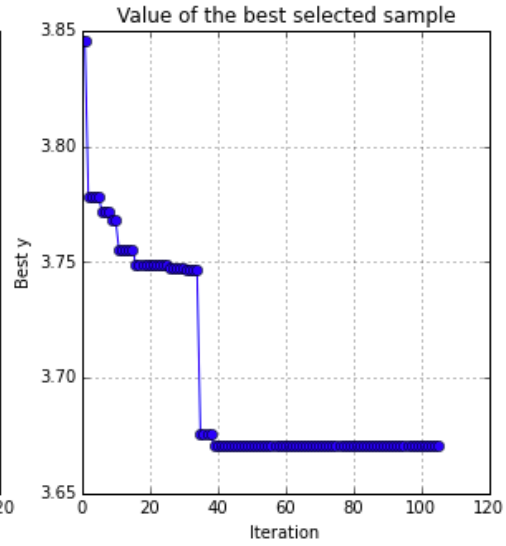
Left=$p(\theta)$, middle= $E(\theta)$, right = $\tilde{J}(\theta)$

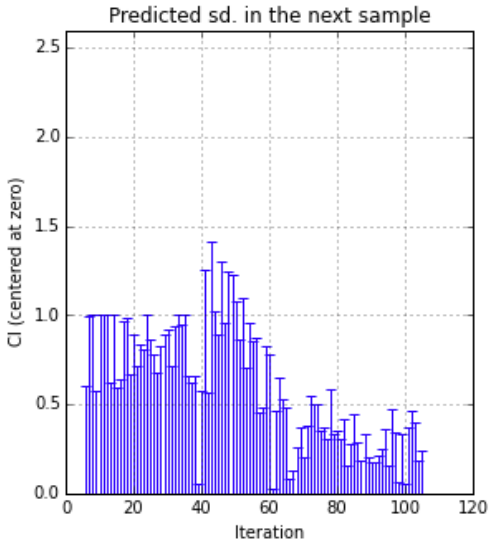




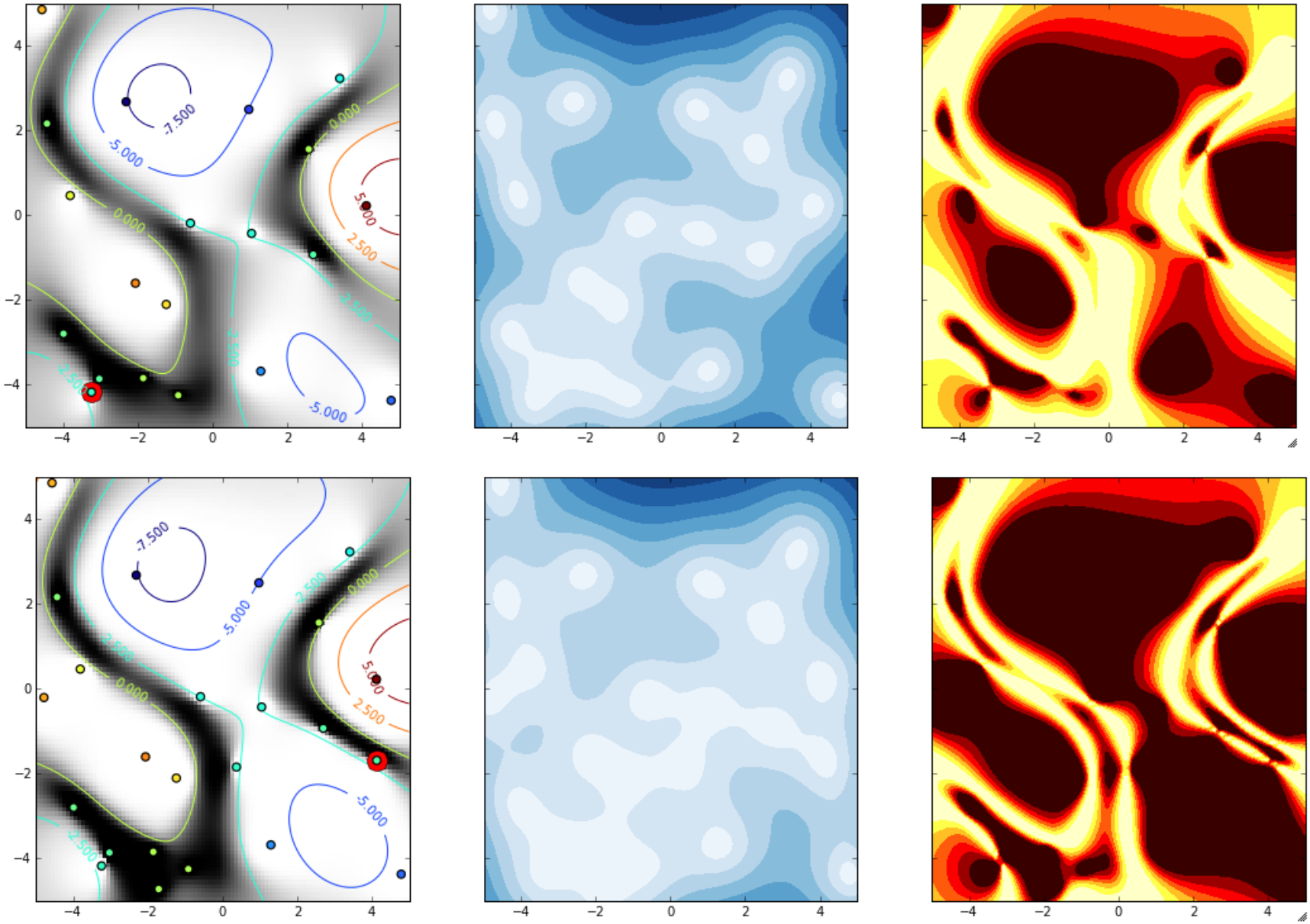








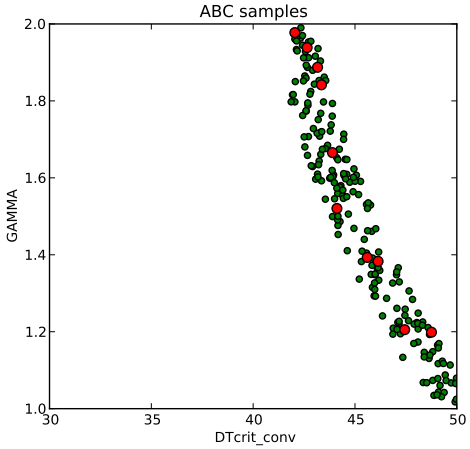
Iterations 20 and 24

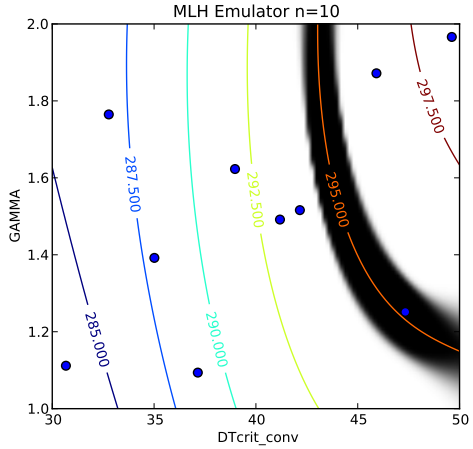


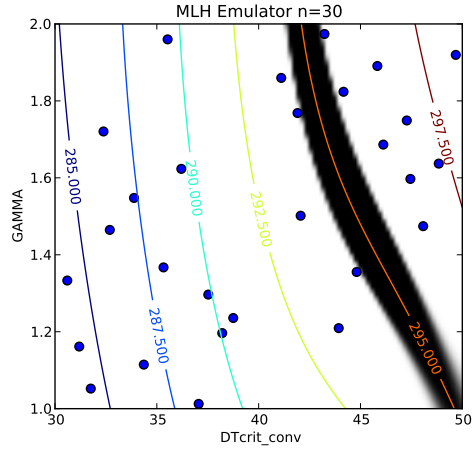
Video

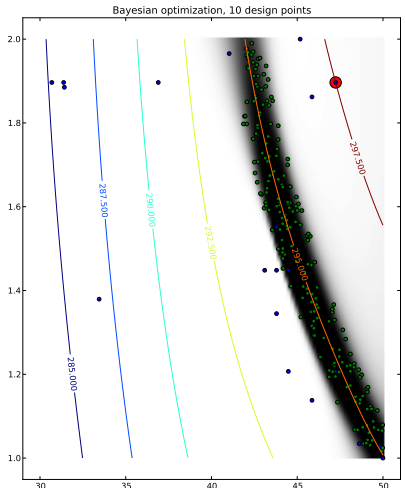
EPm: climate model

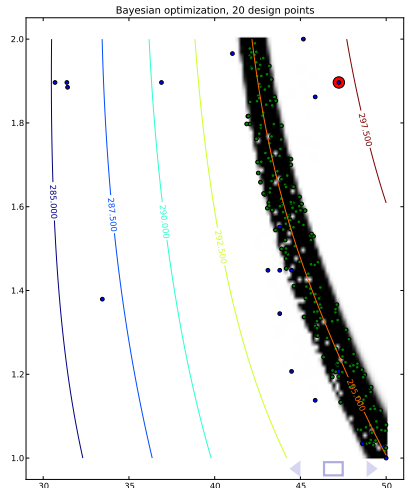
- 3d problem
- DTcrit_conv - critical temperature gradient that triggers convection
- GAMMA - emissivity parameter for water vapour
- Calibrate to global average surface temperature











Difficulties

- Currently, each wave considered in turn. Classification errors in earlier waves can never be corrected. Is it possible to use a design criterion that operates across waves?
- We have to estimate GP hyper-parameters. Designs optimal for history-matching, are non-optimal for GP hyper-parameter estimation - do we need a design that is a trade-off?
- Efficient calculation of the posterior given the final emulator: HMC-NUTS, ...
- How do we marginalise across GP hyper-parameters?
- ⋮

Conclusions

The challenge for ABC is to develop more efficient methods to allow inference in more expensive models.

- Using emulators of the likelihood function, although adding another layer of approximation, does enable progress in hard problems.
- Space-filling designs are inefficient for calibration and history matching problems.
- Entropy based designs give good trade-off between exploration and defining the plausible region
- Bayesian optimisation techniques allow us to solve the hard computation needed to use optimal entropic designs

Conclusions

The challenge for ABC is to develop more efficient methods to allow inference in more expensive models.

- Using emulators of the likelihood function, although adding another layer of approximation, does enable progress in hard problems.
- Space-filling designs are inefficient for calibration and history matching problems.
- Entropy based designs give good trade-off between exploration and defining the plausible region
- Bayesian optimisation techniques allow us to solve the hard computation needed to use optimal entropic designs

Thank you for listening!

r.d.wilkinson@nottingham.ac.uk
www.maths.nottingham.ac.uk/personal/pmzrdw/