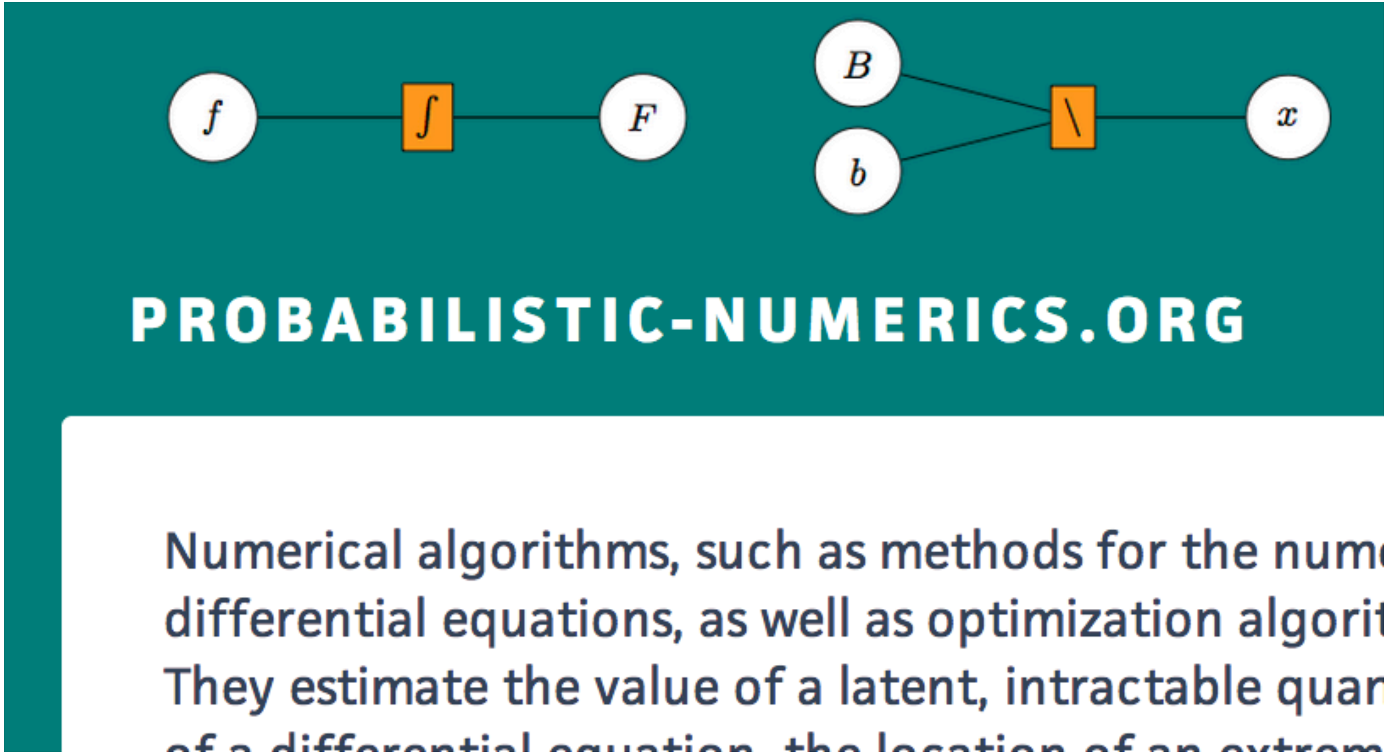
Probabilistic numerics for deep learning

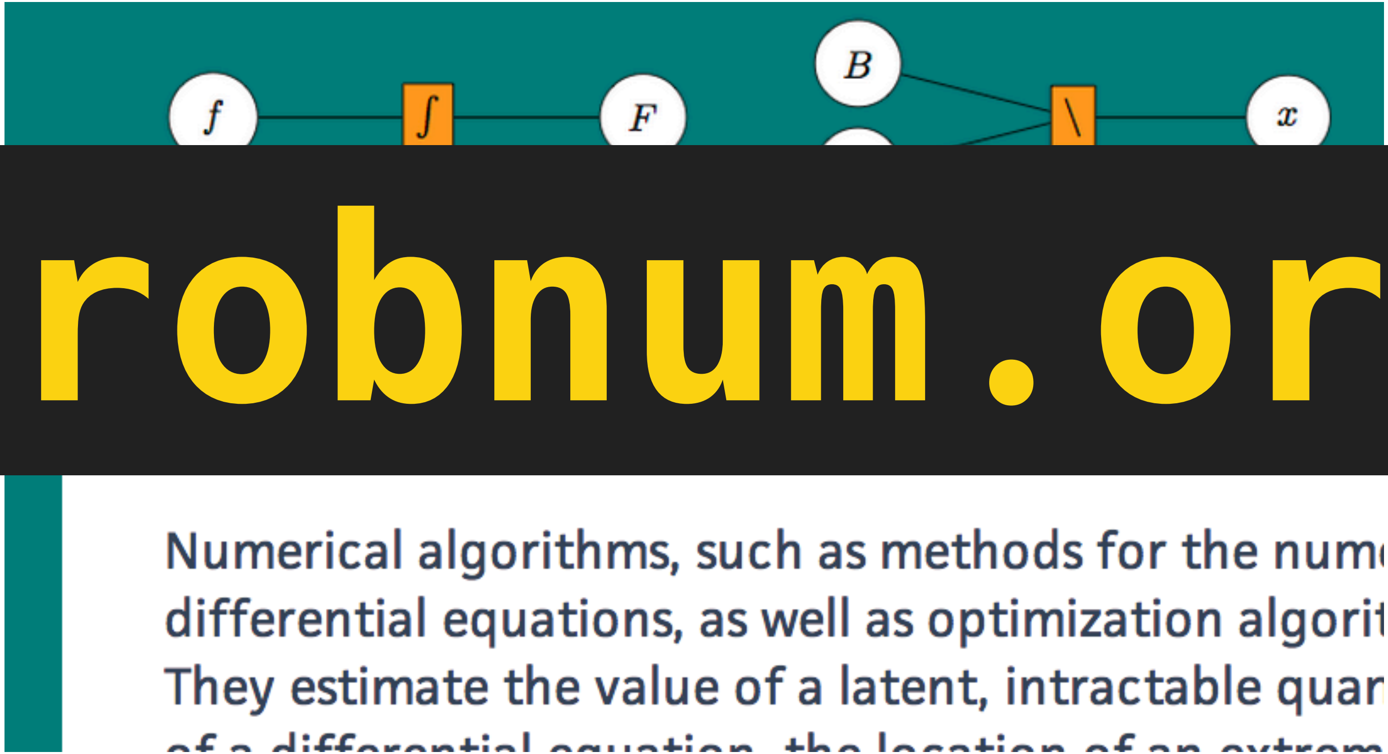
Mike Osborne @maosbot

Philipp Hennig

Probabilistic numerics treats computation as a decision.



Probabilistic numerics treats computation as a decision.



probnum.org

COMMUNITY MEETINGS AND EVENTS

This page lists past and future meetings of the Probabilistic Numerics community.

2017

- *June 18 - 23*
  Dobbiaco Summer School on Probabilistic Numerics at the Hotel Union in Dobbiaco, Italy.
  Organized by Alfredo Bellen, Stefano Maset and Marino Zennaro (University of Trieste) and Alexander Ostermann (University of Innsbruck).
  Taught by Philipp Hennig & Mark Girolami
- *June 5 - 9*
  Seminar on Probabilistic Scientific Computing: Statistical inference approaches to numerical analysis and algorithm design
  at ICERM (the Institute for Computational and Experimental Research in Mathematics), Brown University, Providence, Rhode Island.
  Organized by Philipp Hennig, George Em Karniadakis, Michael A Osborne, Houman Owhadi and Paris Perdikaris

2016

- *18 August*
  Probabilistic Numerics @ MCQMC 2016
  at Stanford University, California
  organized by Mark Girolami and François-Xavier Briol
- *7 January*
  Probabilistic Numerics: Integrating Inference With Integration @ MCMSki
  in Lenzerheide, Switzerland
  organized by Michael Osborne, Chris Oates and François-Xavier Briol

2015

Probabilistic numerics is the study of numeric methods as learning algorithms.

PROBABILISTIC-NUMERICS.ORG

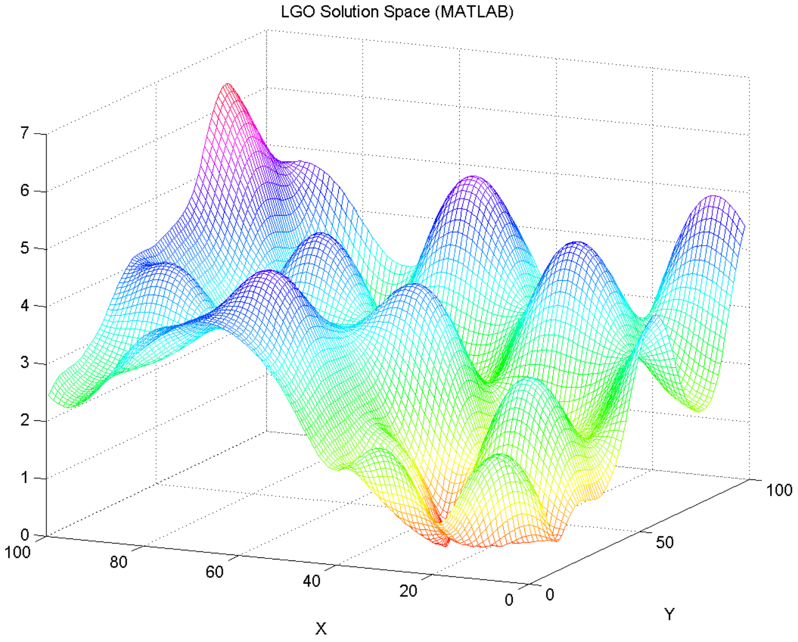
LITERATURE

This page collects literature on all areas of probab
not hesitate to contact us. The fastest way to get
file in /_bibliography, then either send us a pull-re

QUICK-JUMP LINKS:

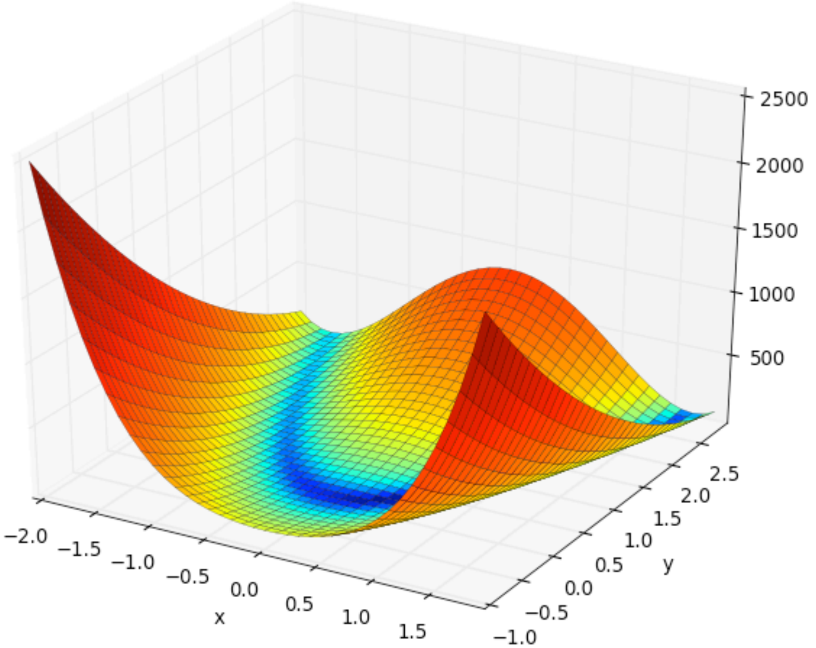
- General and Foundational
- Quadrature
- Linear Algebra
- Optimization
- Ordinary Differential Equations
- Partial Differential Equations

Global optimisation considers objective functions that are multi-modal and often expensive to evaluate.

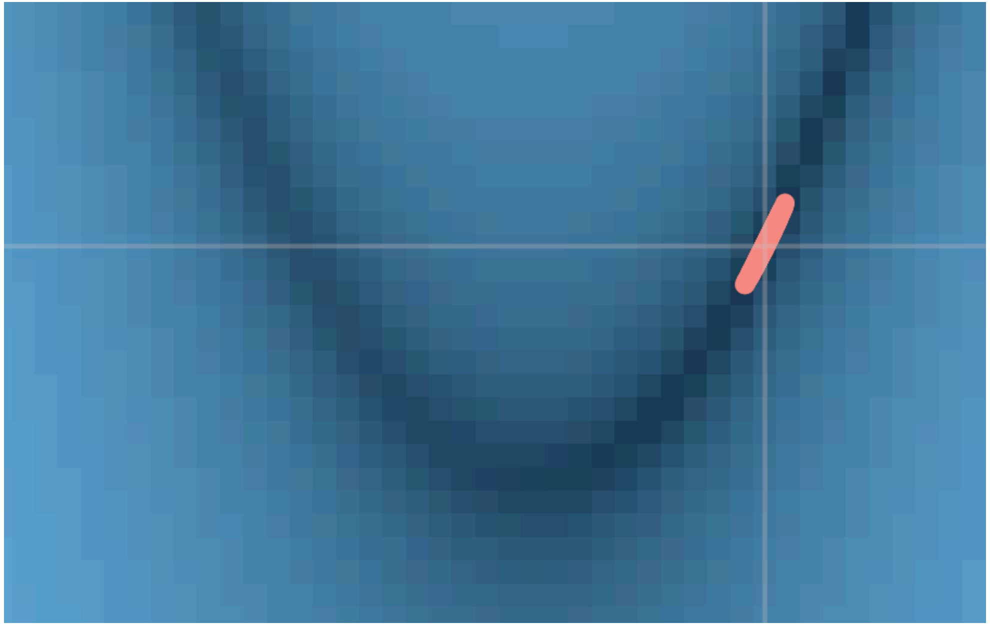


The Rosenbrock is expressible in closed-form.

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

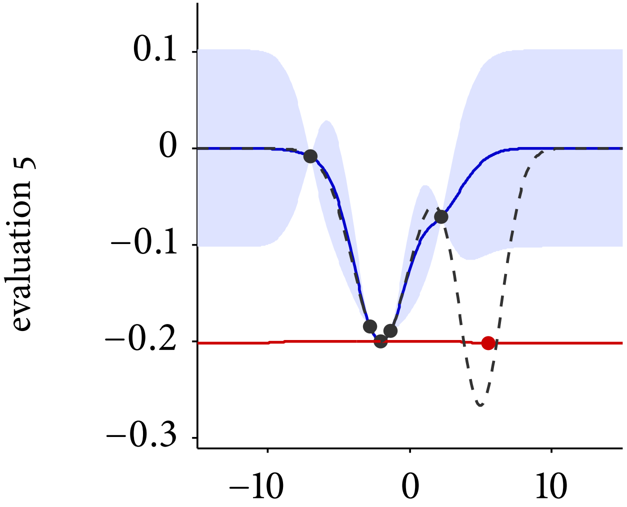


Computational limits form the core of the optimisation problem.



$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

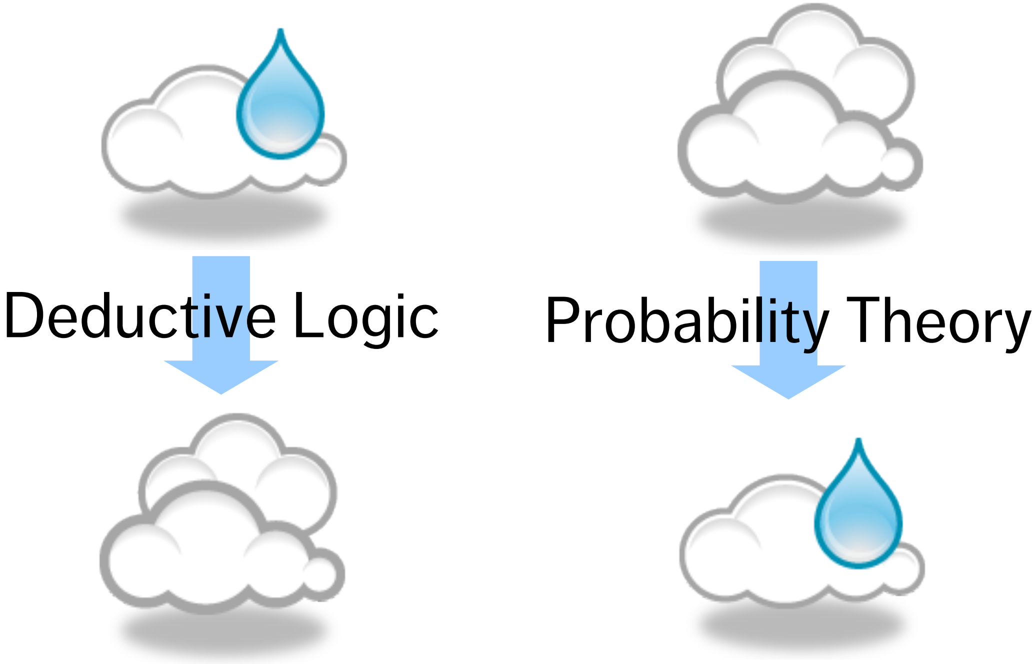
We are epistemically uncertain about *f(x,y)* due to being unable to afford its computation.



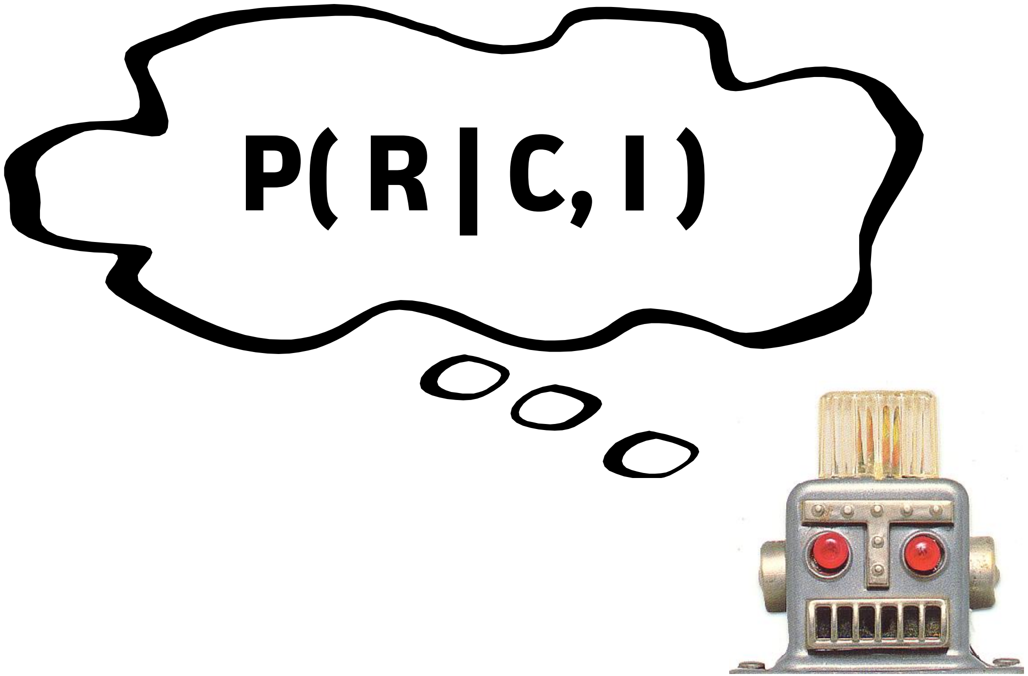
We can hence probabilistically model *f(x,y)*, and use decision theory to make optimal use of computation.

Probabilistic modelling of functions

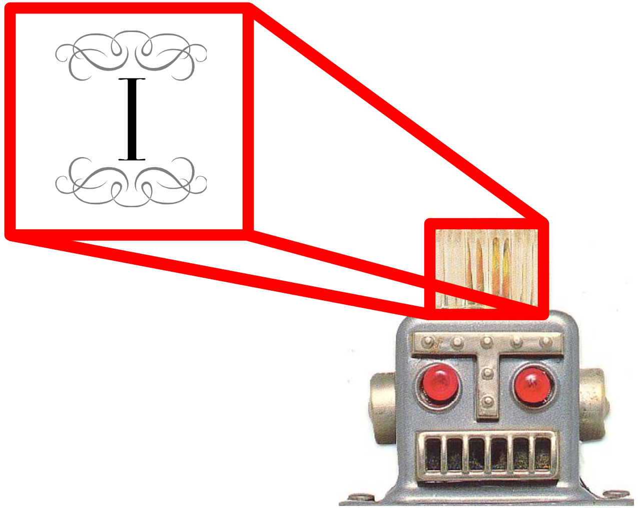
Probability theory represents an extension of traditional logic, allowing us to reason in the face of uncertainty.



A probability is a degree of belief. This might be held by any agent – a human, a robot, a pigeon, etc.

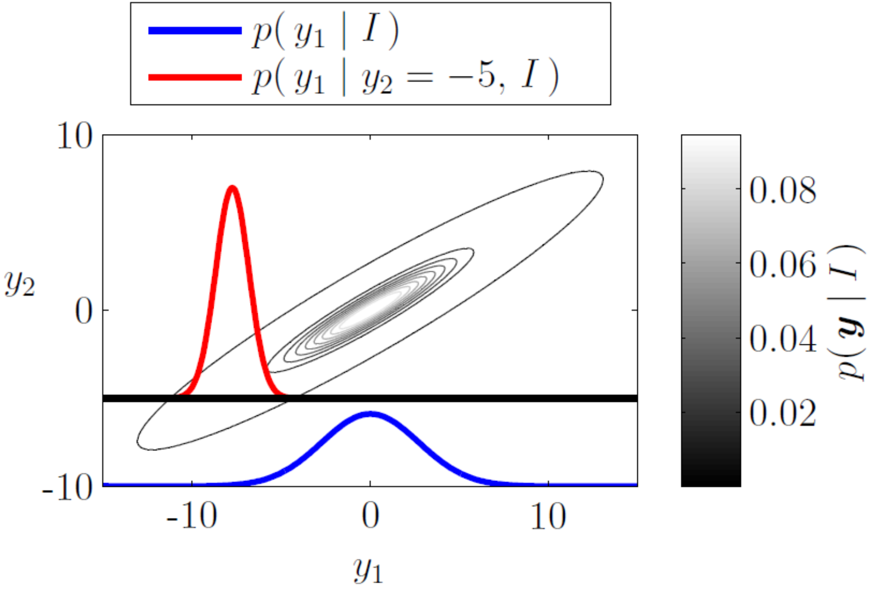


‘I’ is the totality of an agent’s prior information. An agent is (partially) defined by I.

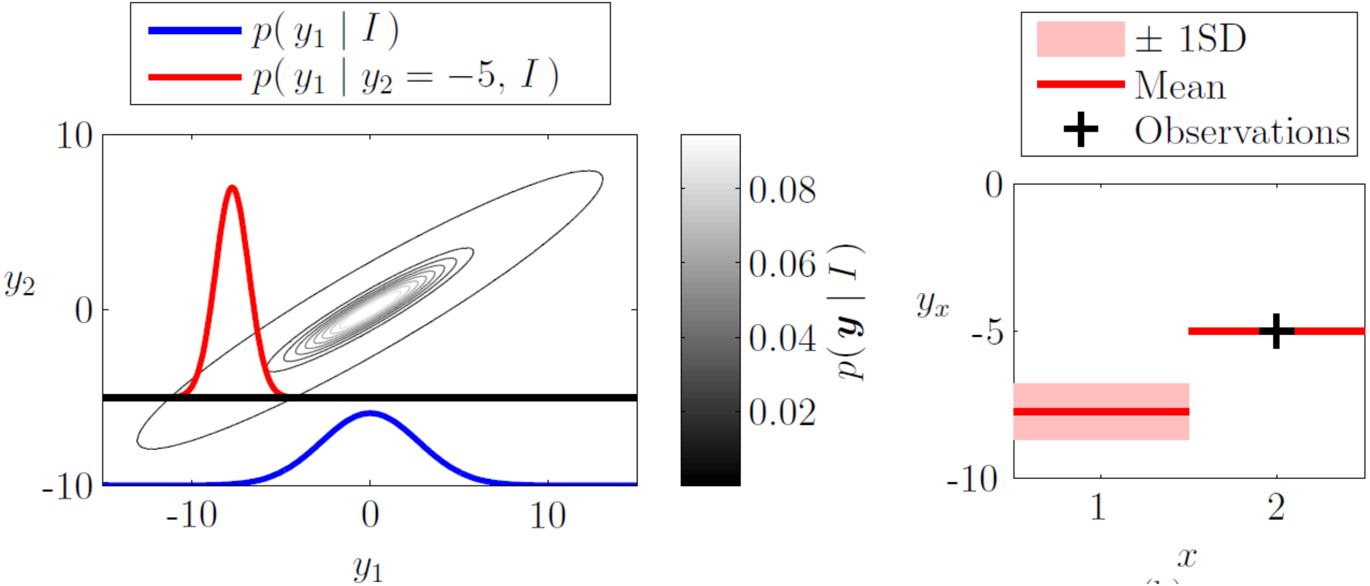


We define our agents so that they can perform difficult inference for us.

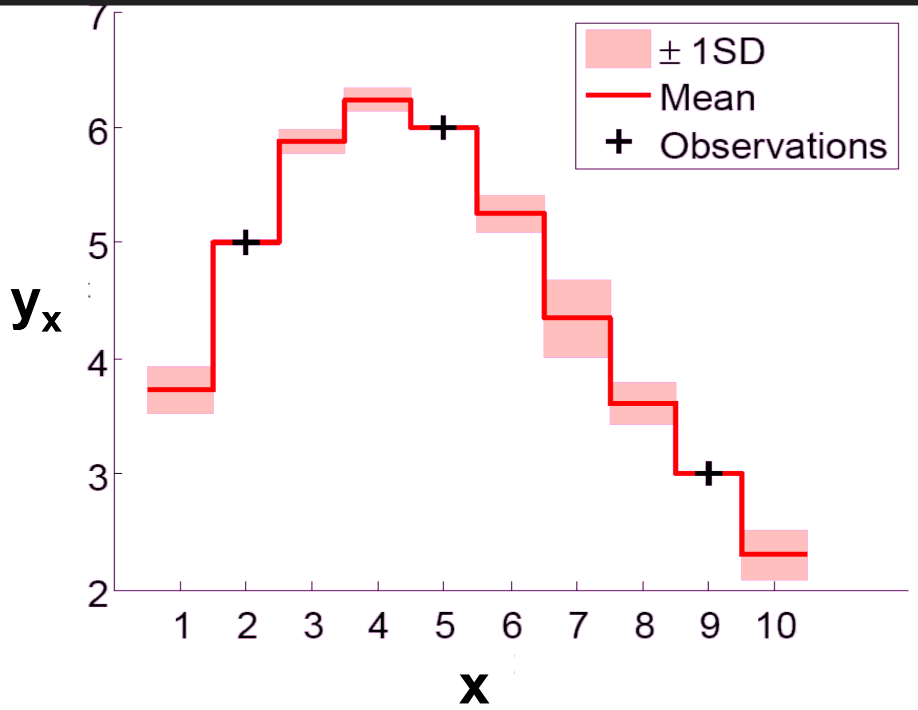
The Gaussian distribution allows us to produce distributions for variables conditioned on any other observed variables.



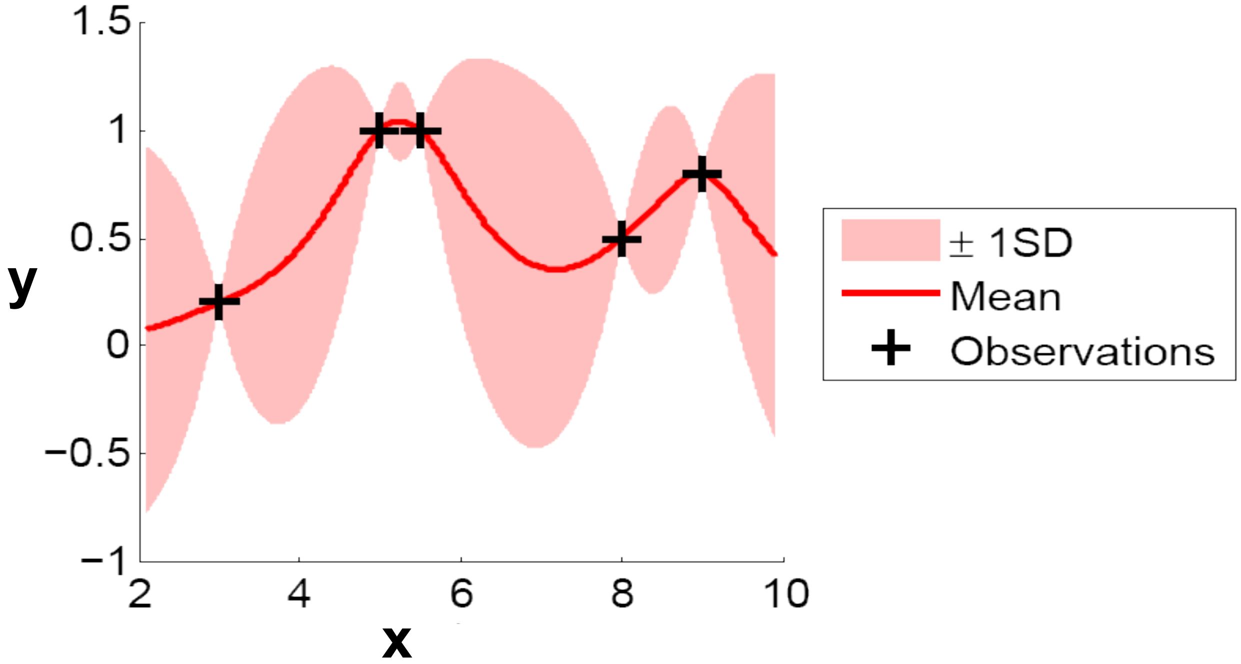
The Gaussian distribution allows us to produce distributions for variables conditioned on any other observed variables.



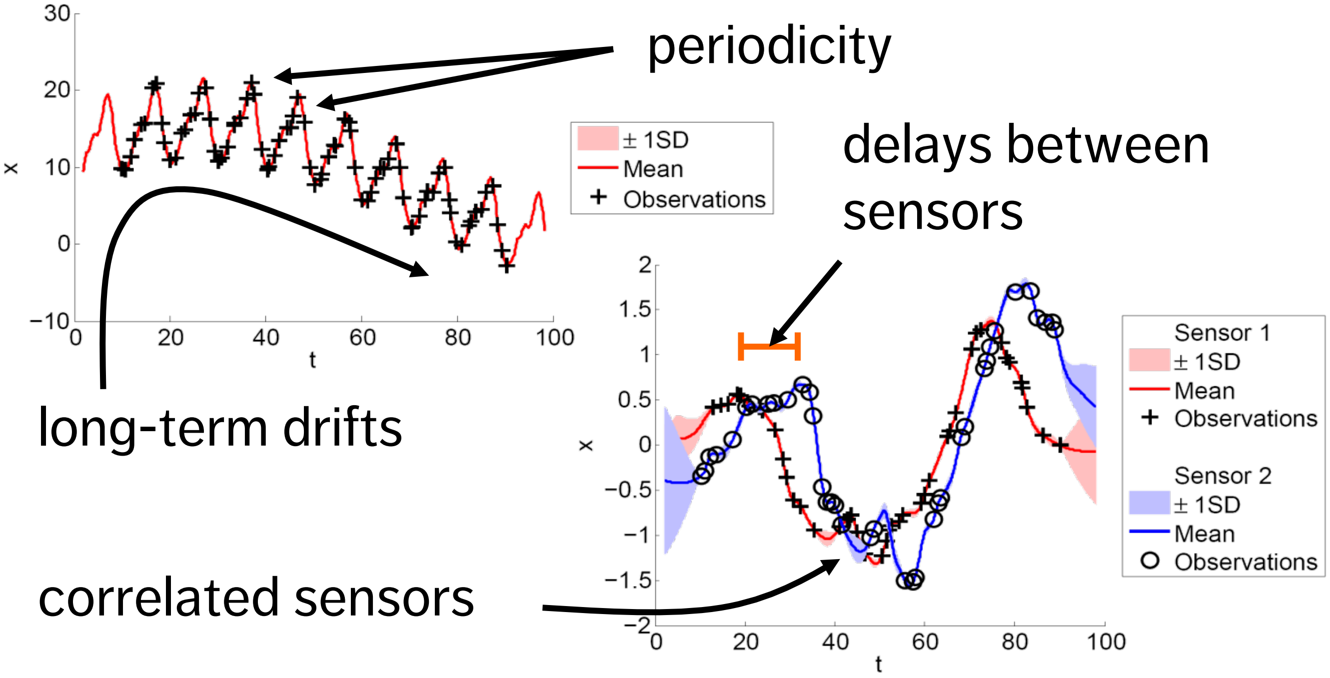
A Gaussian process is the generalisation of a multivariate Gaussian distribution to a potentially infinite number of variables.



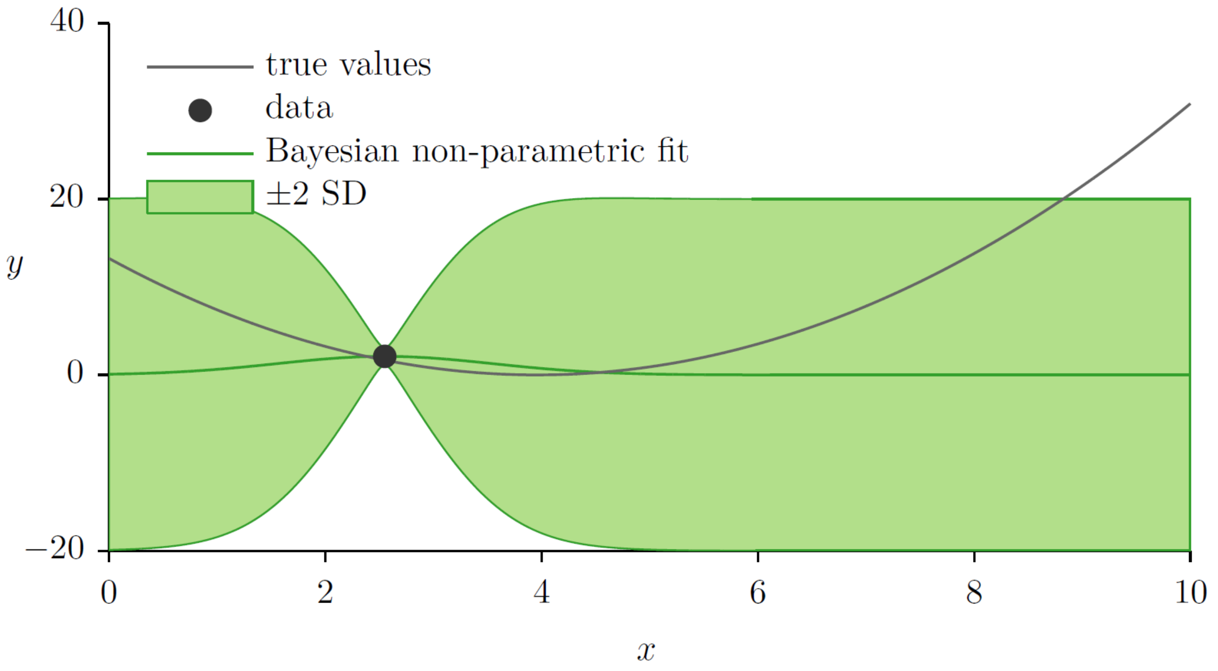
A Gaussian process provides a non-parametric model for functions, defined by mean and covariance functions.

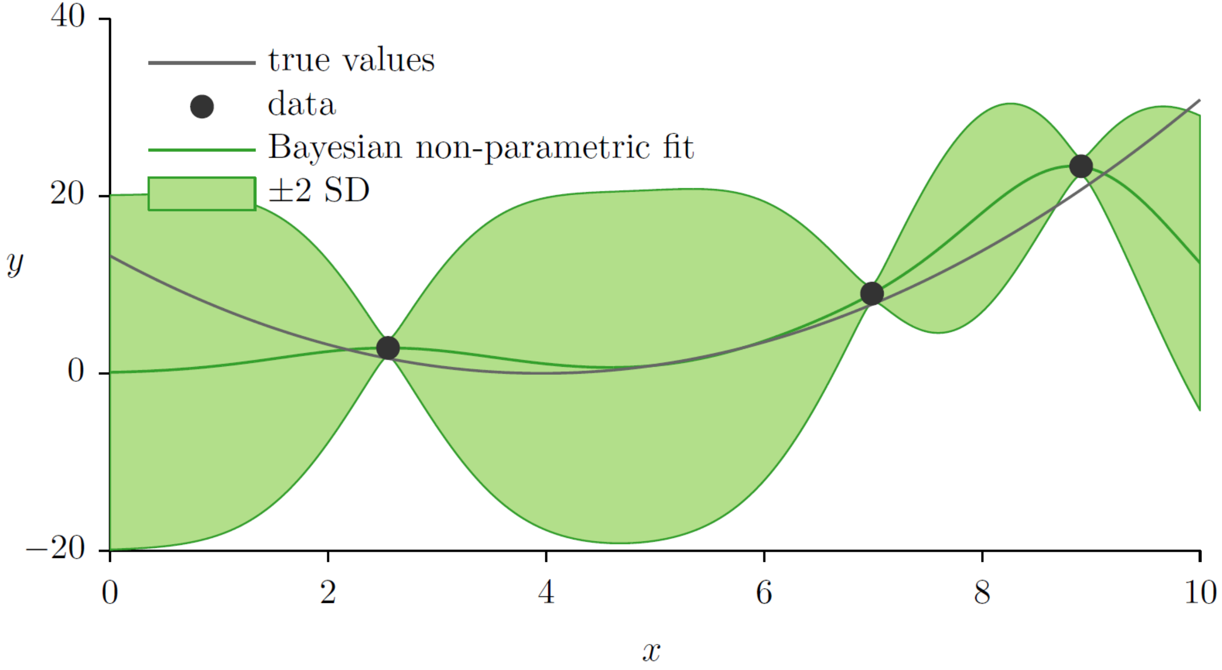


Gaussian processes are specified by a covariance function, which flexibly allow the expression of e.g.

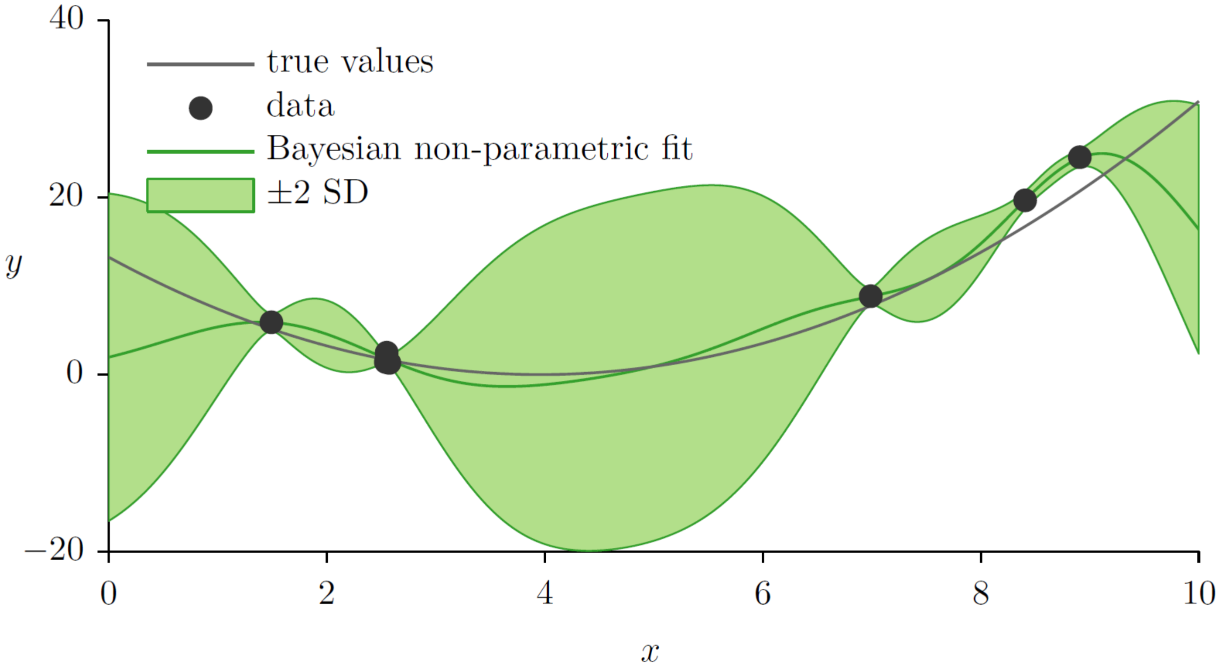


Gaussian processes have a complexity that grows with the data; they provide flexible models, robust to overfitting.

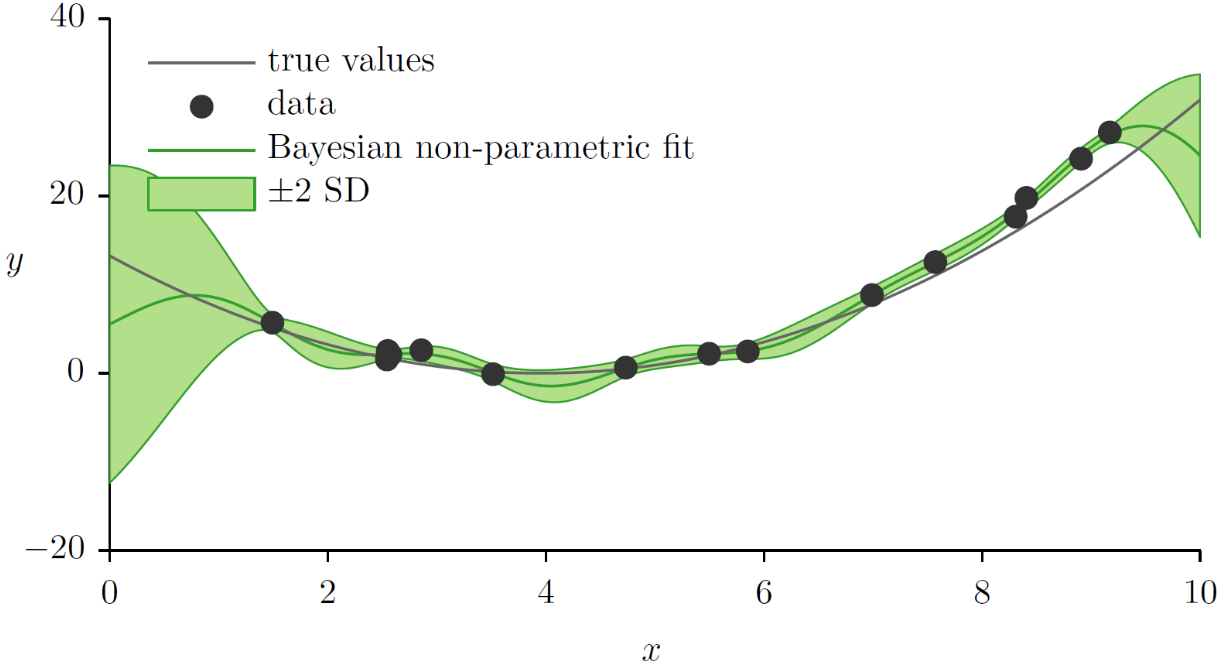




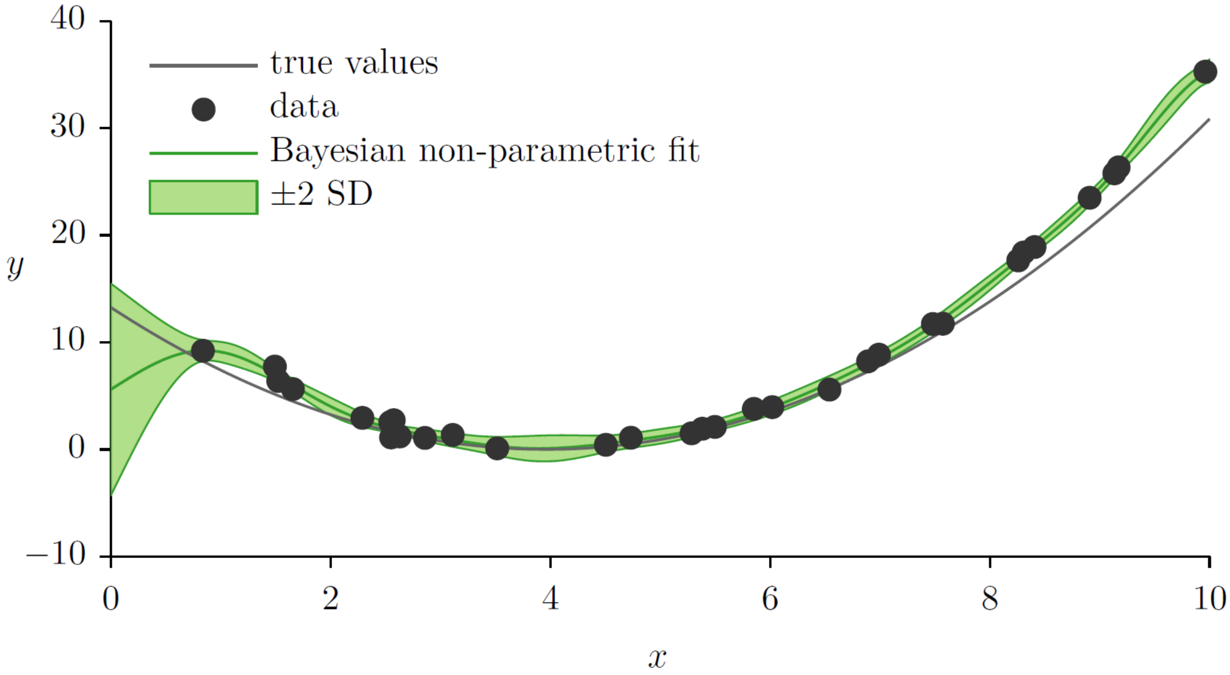
40
20
0
−20
y
0
2
4
6
8
10
x
true values
data
Bayesian non-parametric fit
±2 SD



40
20
0
−20
y
0
2
4
6
8
10
x
true values
data
Bayesian non-parametric fit
±2 SD



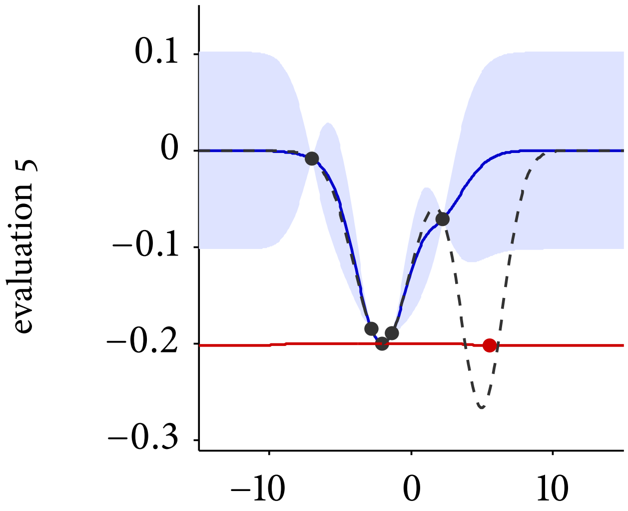
40
20
0
−20
y
0
2
4
6
8
10
x
true values
data
Bayesian non-parametric fit
±2 SD



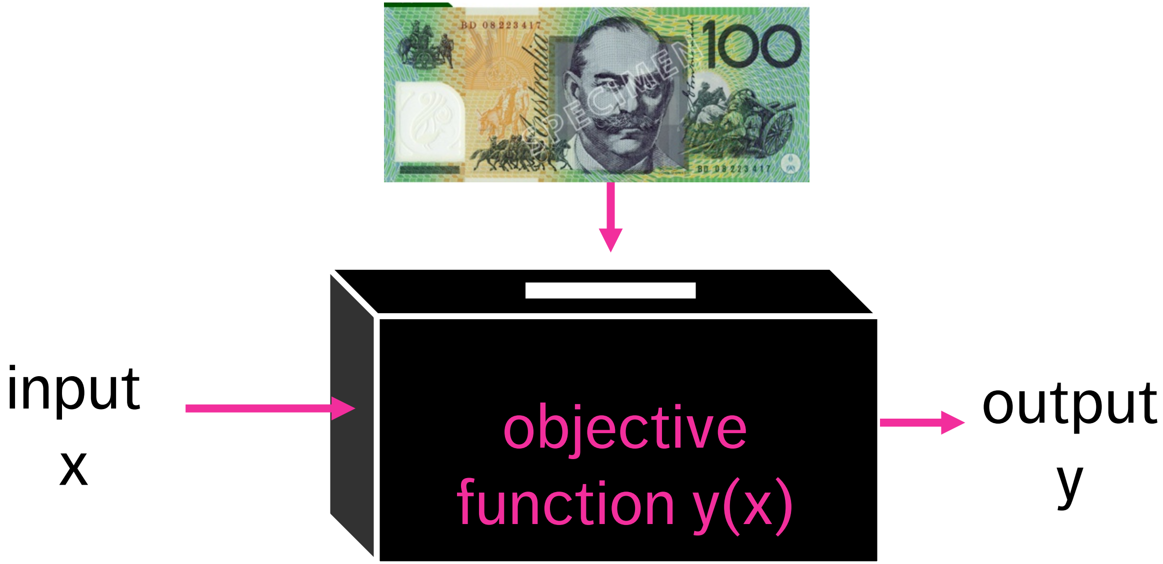
true values
data
Bayesian non-parametric fit
±2 SD
40
30
20
10
0
−10
y
0
2
4
6
8
10
x

Bayesian optimisation as decision theory

Bayesian optimisation is the approach of probabilistically modelling *f(x,y)*, and using decision theory to make optimal use of computation.



By defining the costs of observation and uncertainty, we can select evaluations optimally by minimising the expected loss with respect to a probability distribution.



We define a loss function that is the lowest function value found after our algorithm ends.

Assuming that we have only one evaluation remaining, the loss of it returning value y, given that the current lowest value obtained is η, is

$$\lambda(y) \triangleq \begin{cases} y; & y < \eta \\ \eta; & y \geq \eta \end{cases}.$$

This loss function makes computing the expected loss simple: we'll take a myopic approximation and consider only the next evaluation.

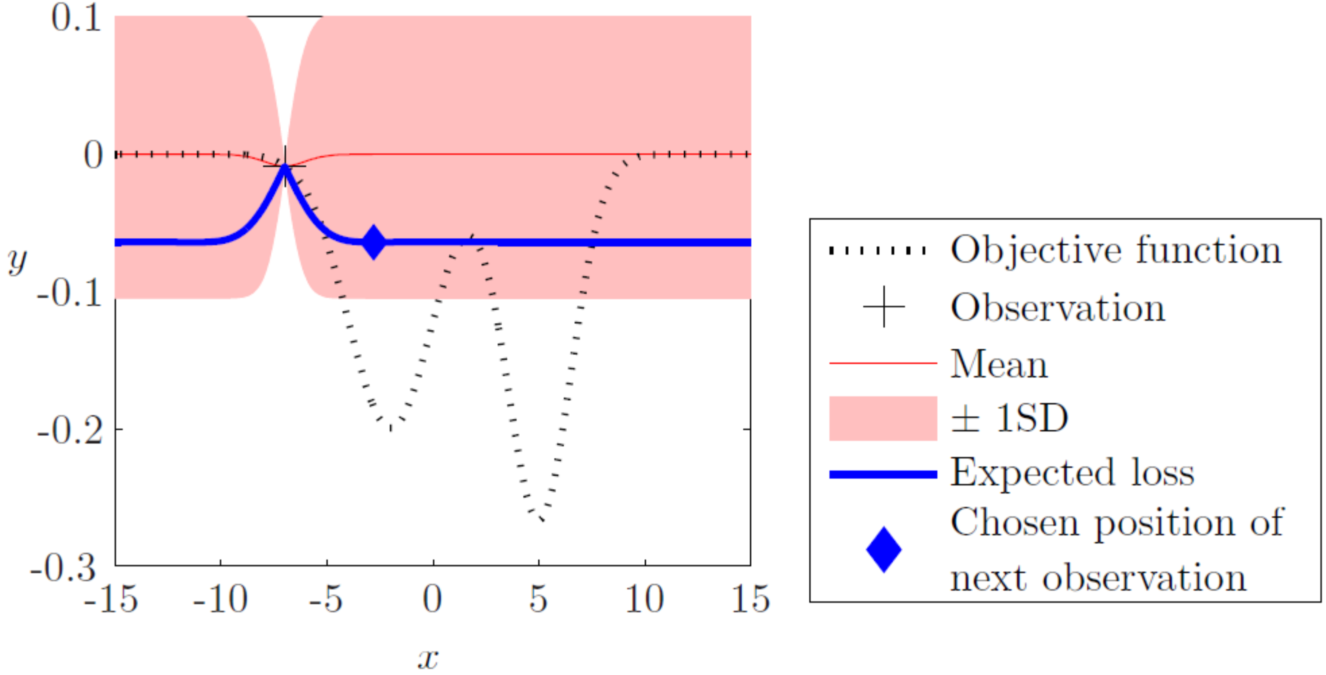
$$\int \lambda(y)\, p(\, y \mid x,\, I_0\,)\ \mathrm{d}y$$

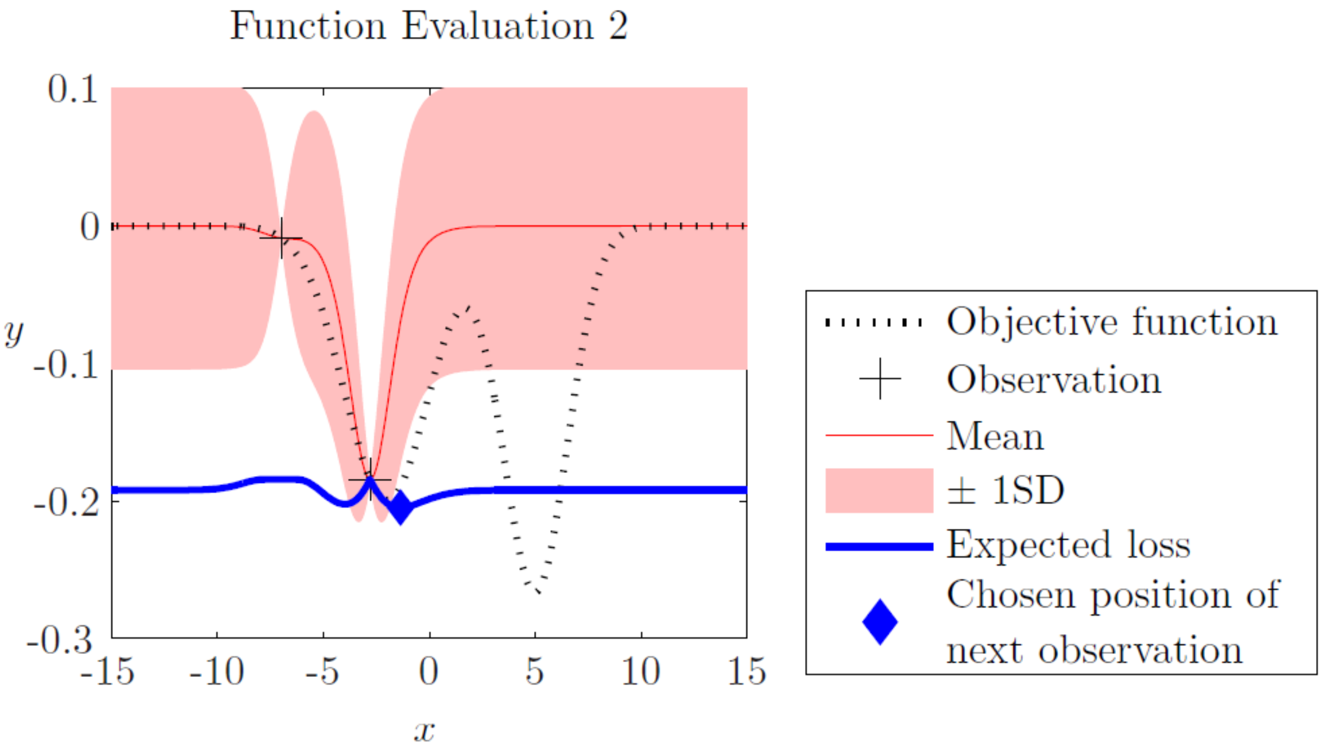
I_0 : All available information.

x : Next evaluation location.

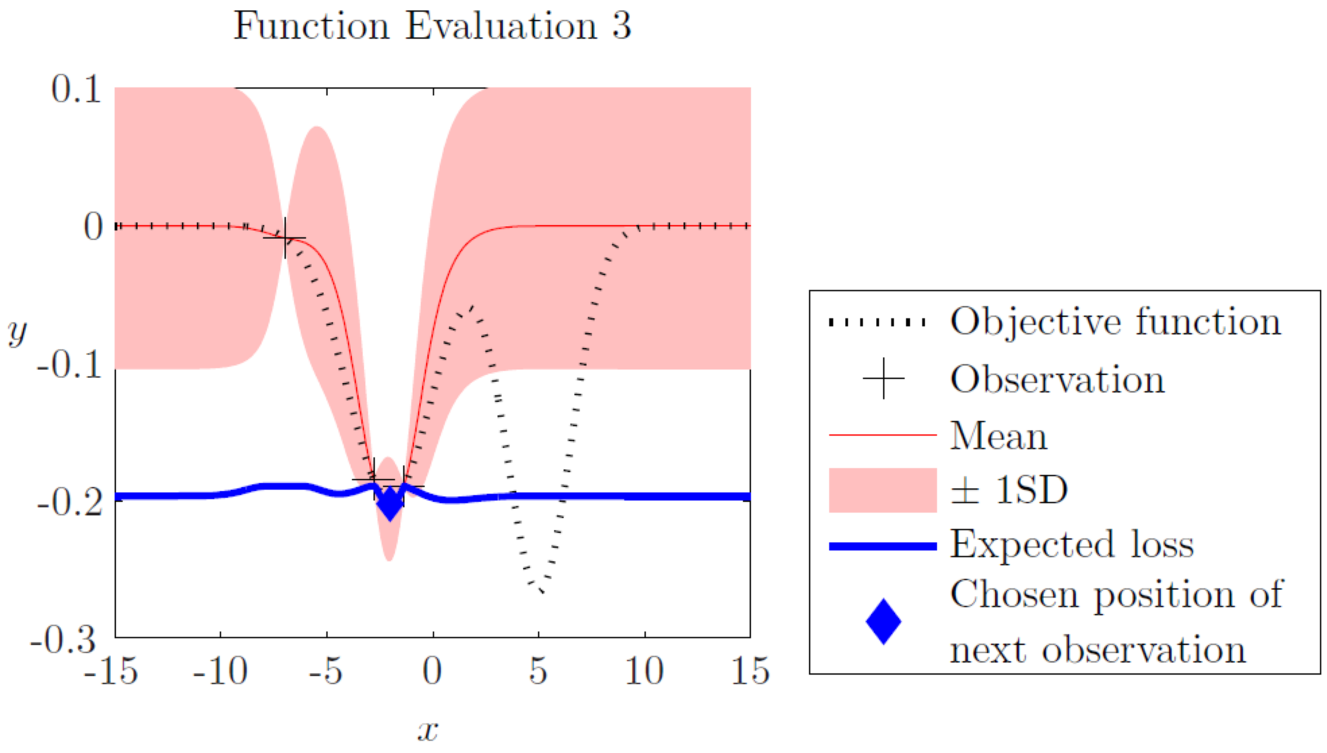
The expected loss is the expected lowest value of the function we've evaluated after the next evaluation.

We choose a Gaussian process as the probability distribution for the objective function, giving a tractable expected loss.

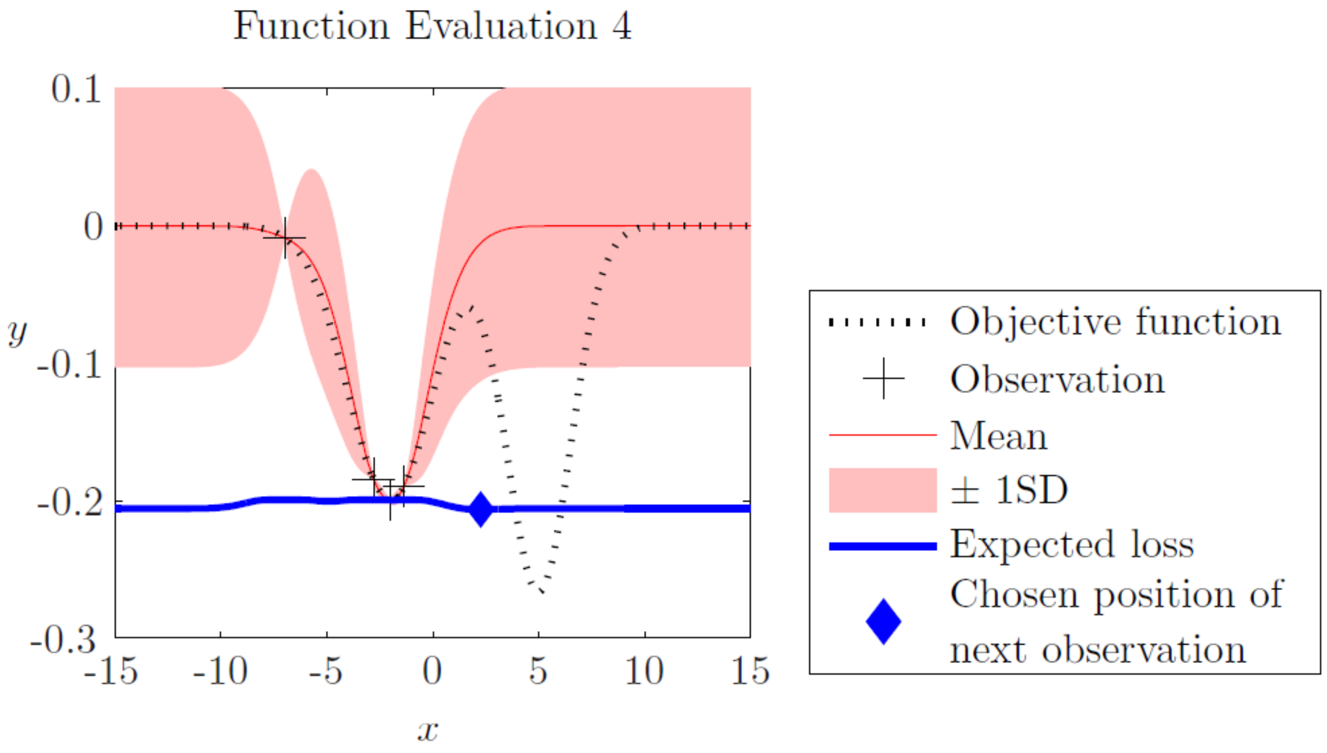


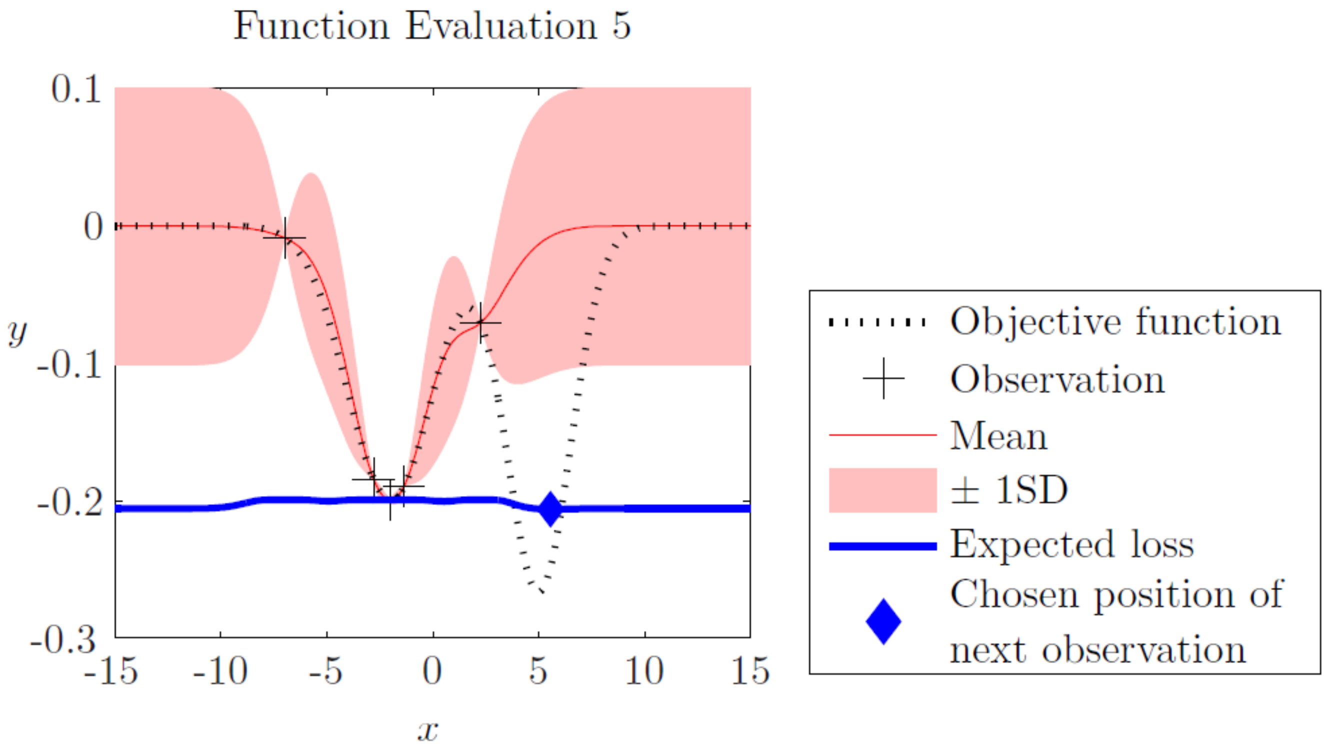


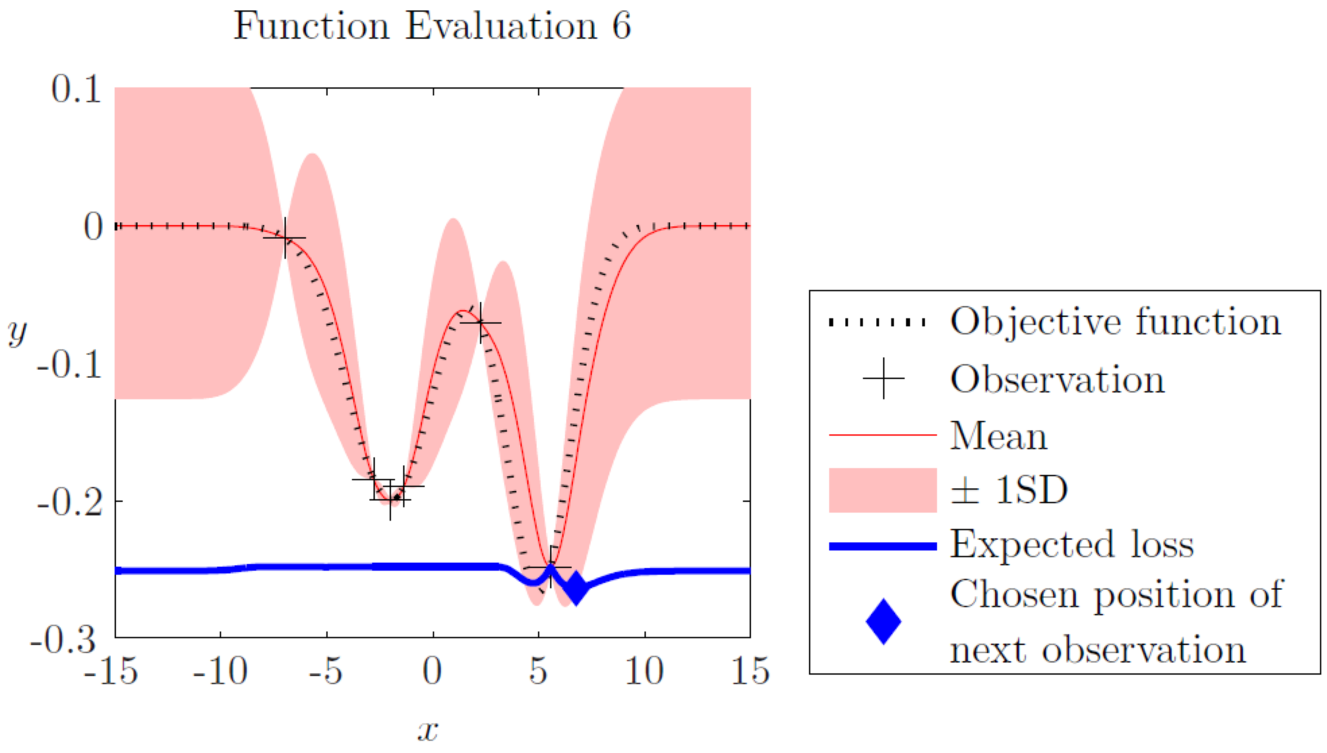
Function Evaluation 2
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation

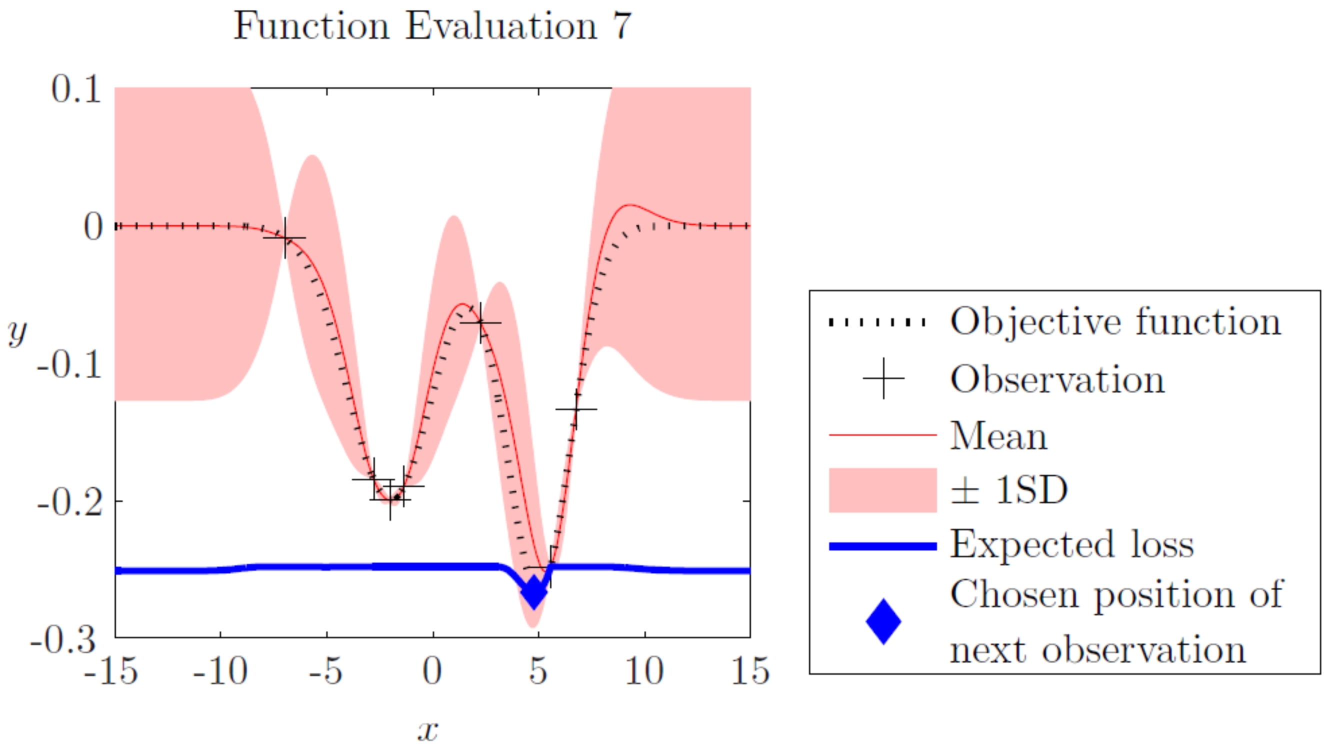


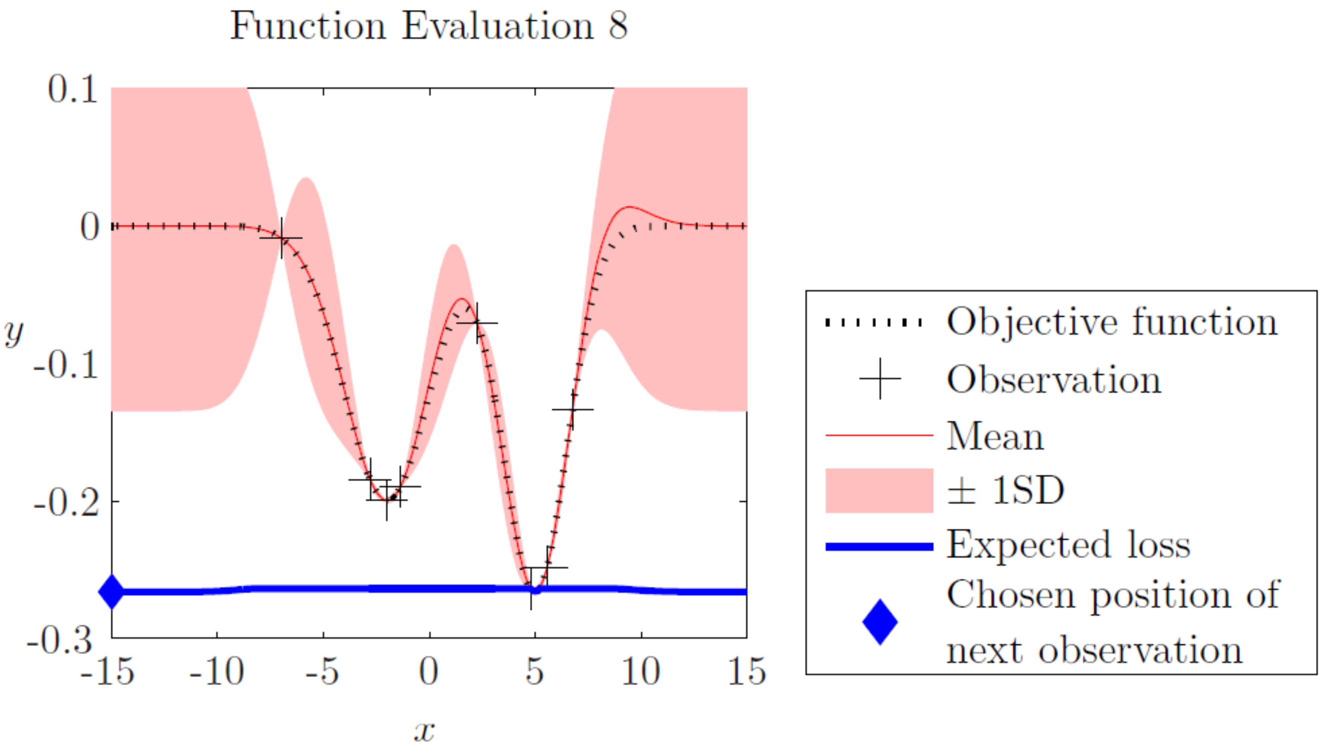
Function Evaluation 3
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation

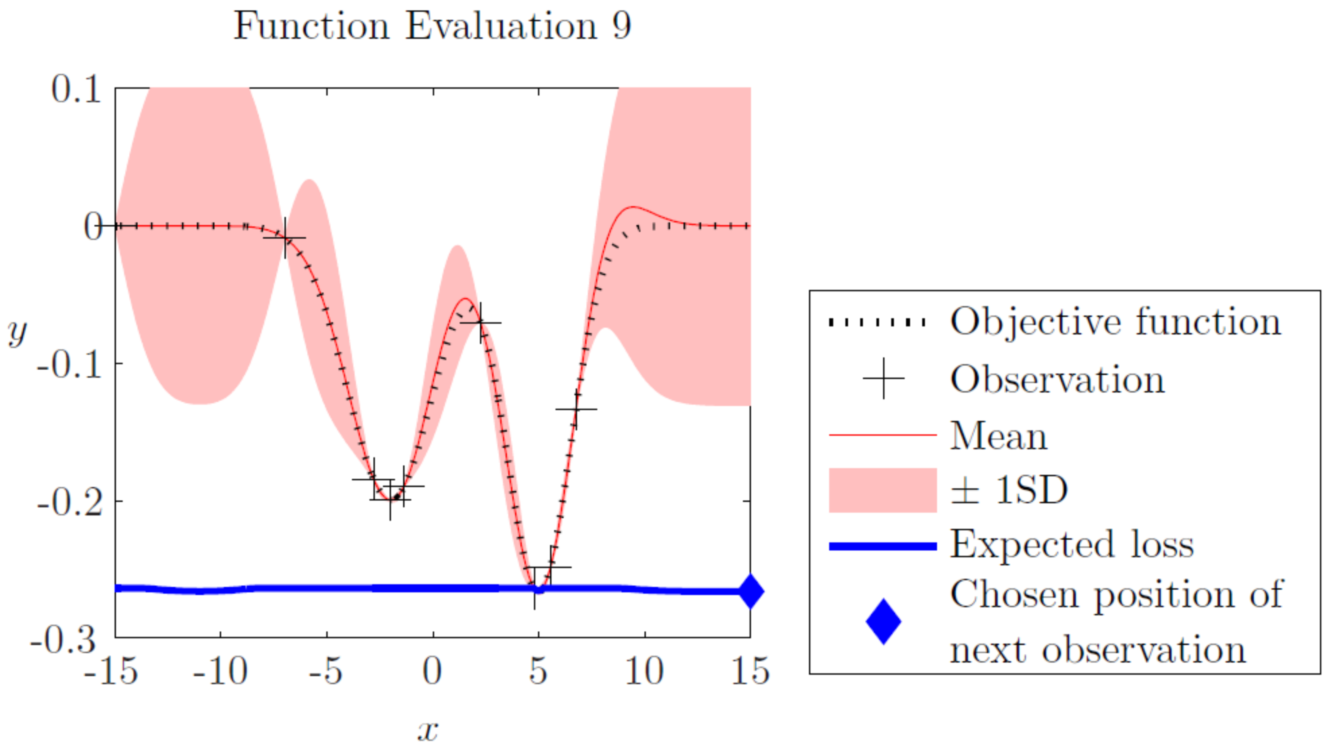

Function Evaluation 4
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation


Function Evaluation 5
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation


Function Evaluation 6
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation


Function Evaluation 7
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of
next observation

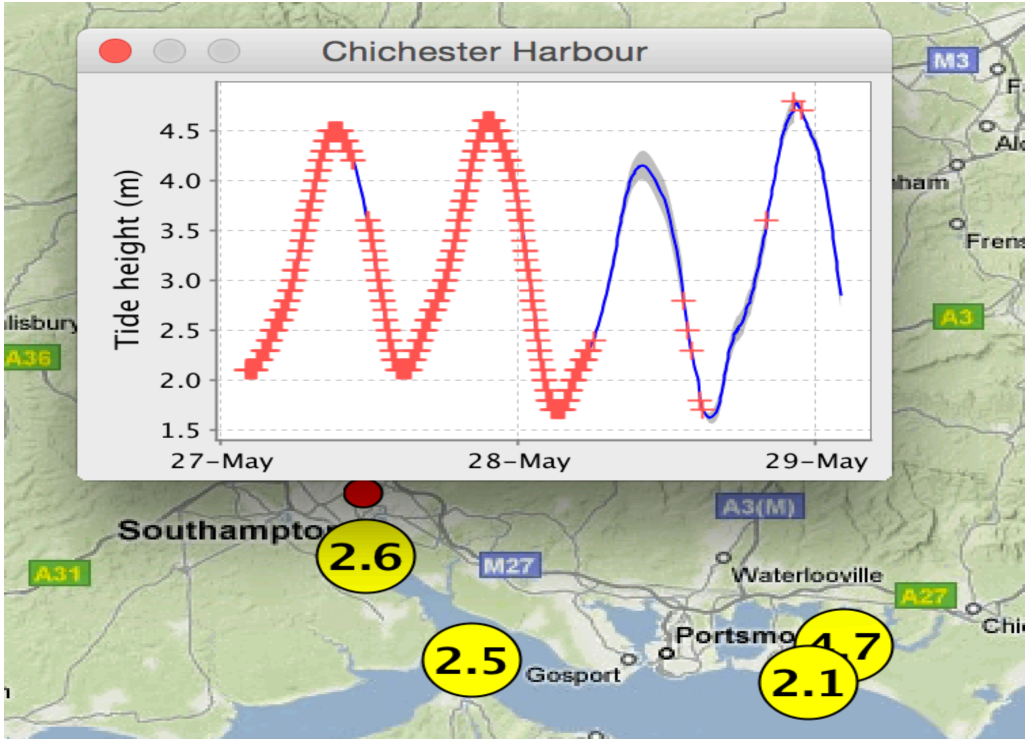

Function Evaluation 8
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation



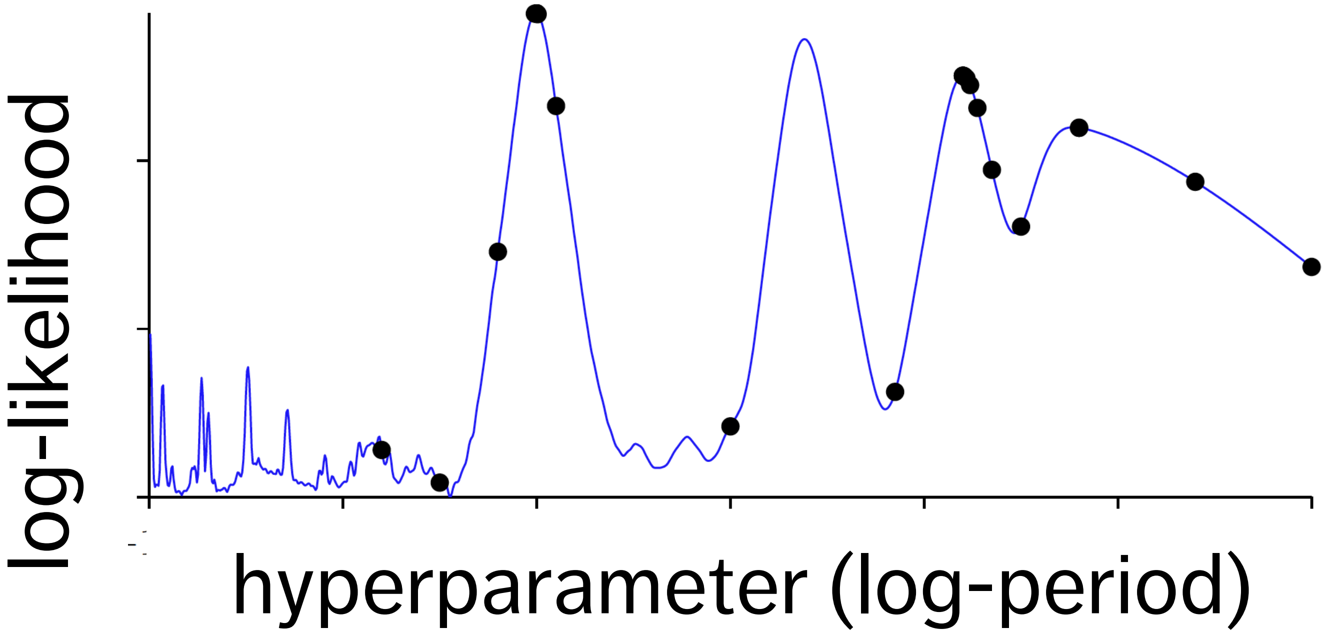
Function Evaluation 9
0.1
0
-0.1
-0.2
-0.3
y
-15
-10
-5
0
5
10
15
x
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen position of next observation

Bayesian optimisation for tuning hyperparameters

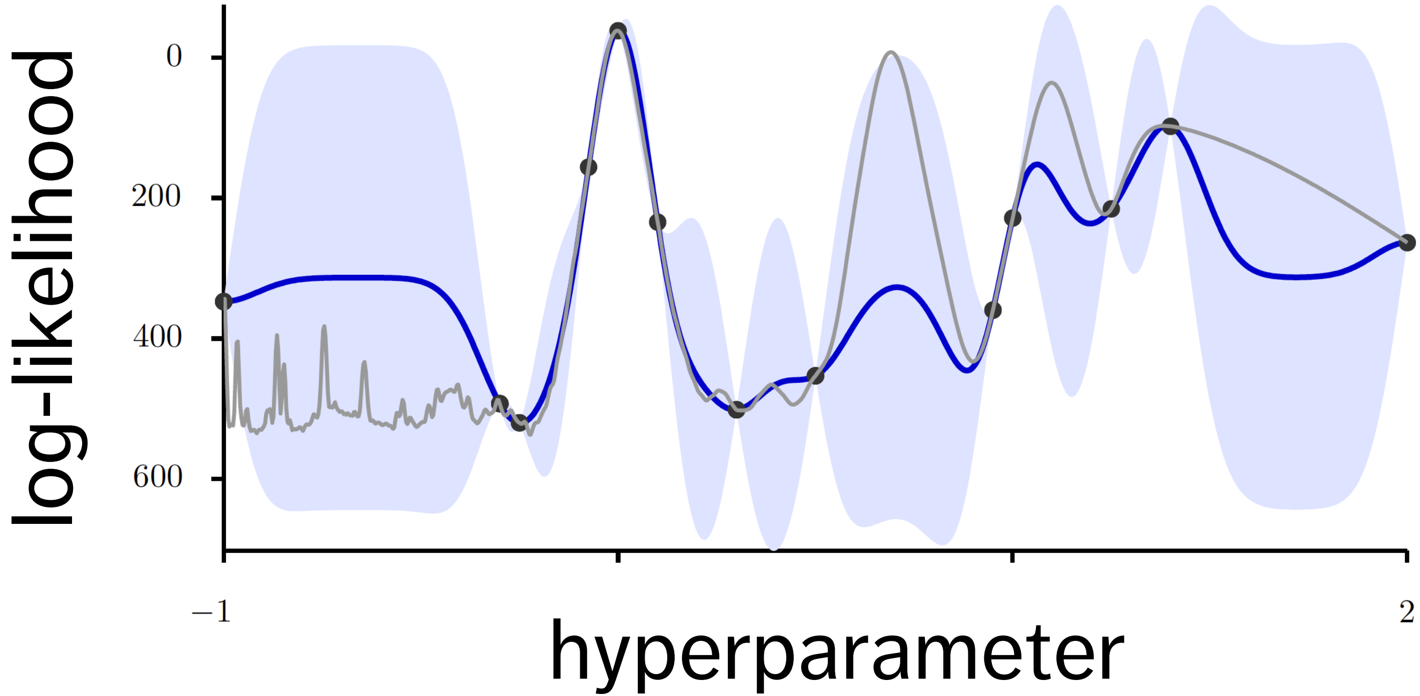
Tuning is used to cope with model parameters (such as periods).



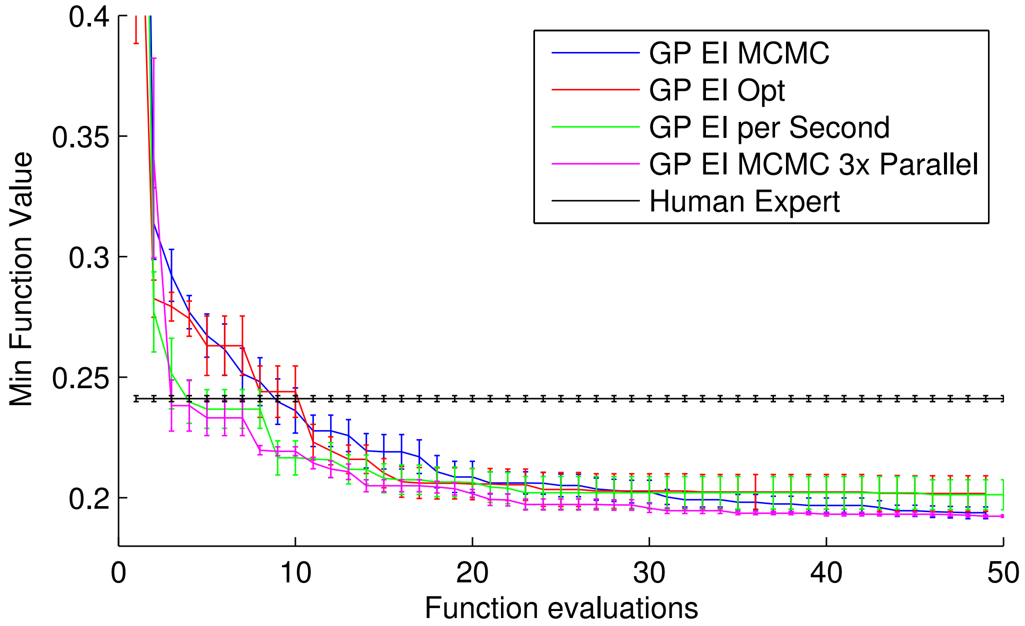
Optimisation (as in maximum likelihood or least squares), gives a reasonable heuristic for exploring the likelihood.



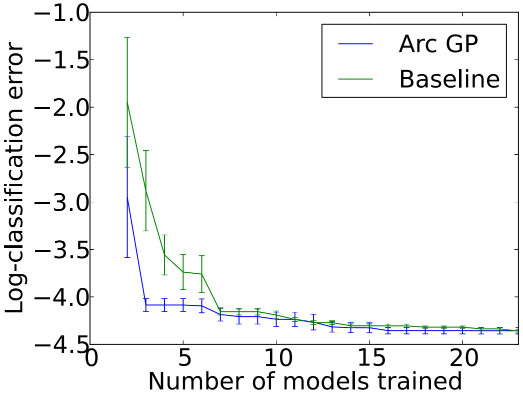
Bayesian optimisation gives a powerful method for such tuning.



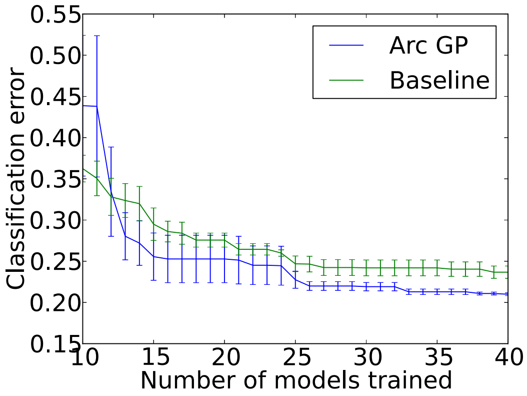
Snoek, Larochelle and Adams (2012) used Bayesian optimisation to tune convolutional neural networks.



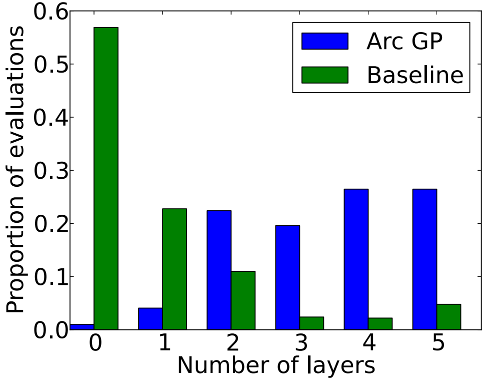
Bayesian optimisation is useful in automating structured search over # hidden layers, learning rates, dropout rates, # hidden units per layer & L2 weight constraints.



(a) MNIST



(b) CIFAR-10

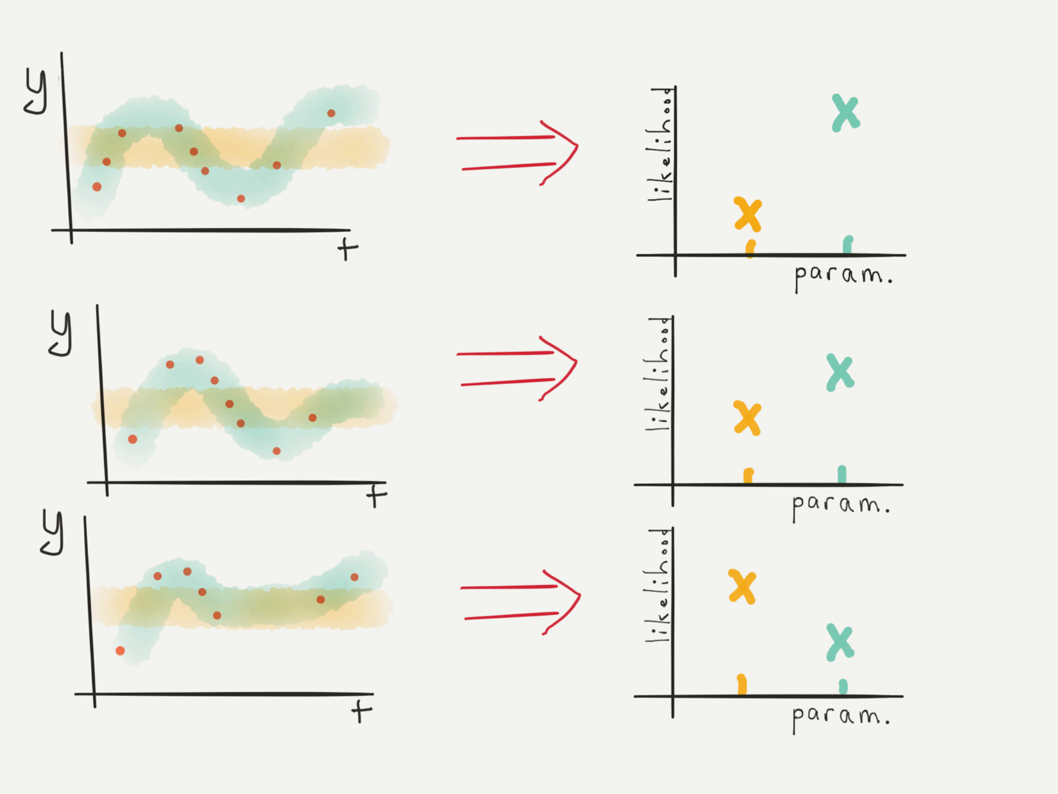


(c) Architectures searched

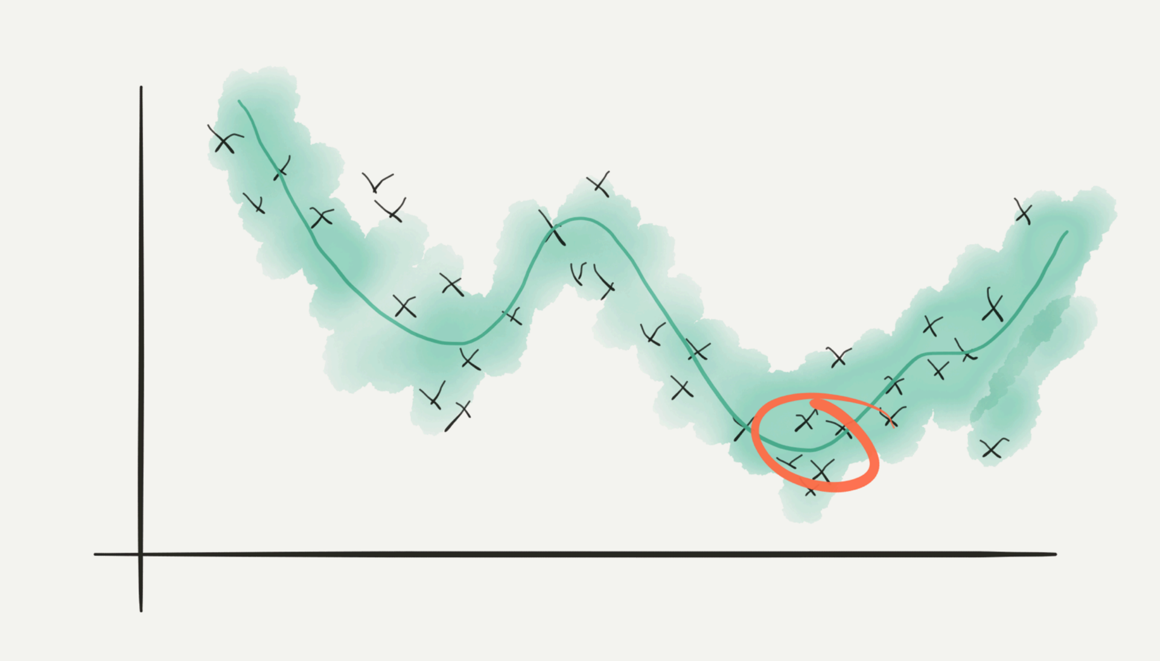
Source: Swersky et al (2013)

Bayesian stochastic optimisation

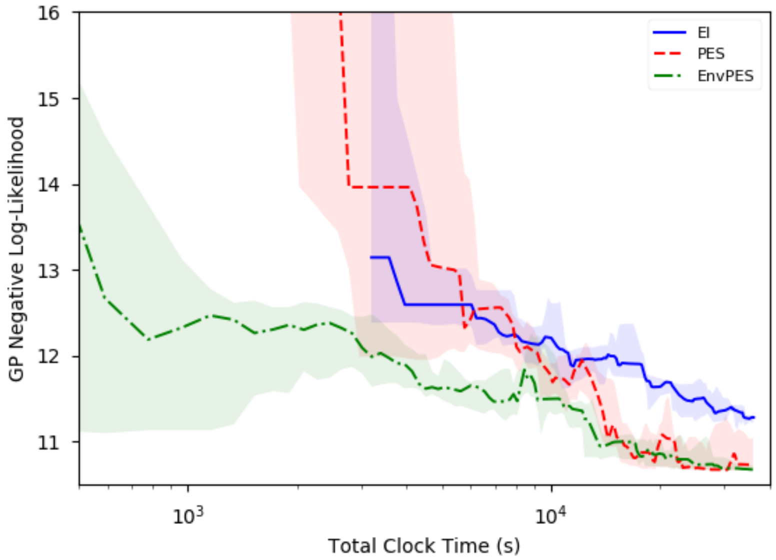
Using only a subset of the data (a mini-batch) gives a noisy likelihood evaluation.



If we use Bayesian optimisation on these noisy evaluations, we can perform stochastic learning.



Lower-variance evaluations (on smaller subsets) are higher cost: let's also Bayesian optimise over the fidelity of our evaluations!



We tune the hyperparameters of a GP fitted to half hourly time series data for UK electricity demand for 2015, for which a full evaluation costs ten minutes.

Klein, Falkner, Bartels, Hennig & Hutter (2017);
McLeod, Osborne & Roberts (2017), `arxiv.org/abs/1703.04335`

Quiz: which of these sequences is random?

1.6224441111111114444443333333

2.1693993751058209749445923078

3.7129042634726105902083360448

4.1000111111101111111001010000

Quiz: which of these sequences is random?

1. 6222444111111114444444333333

   Seven d6 rolls with *i* repeats of the *i*th roll.

2. 1693993751058209749445923078

   The 41st to 70th digits of π.

3. 7129042634726105902083360448

   This sequence was generated by the von Neumann method with seed 908344.

4. 1000111111011111111001010000

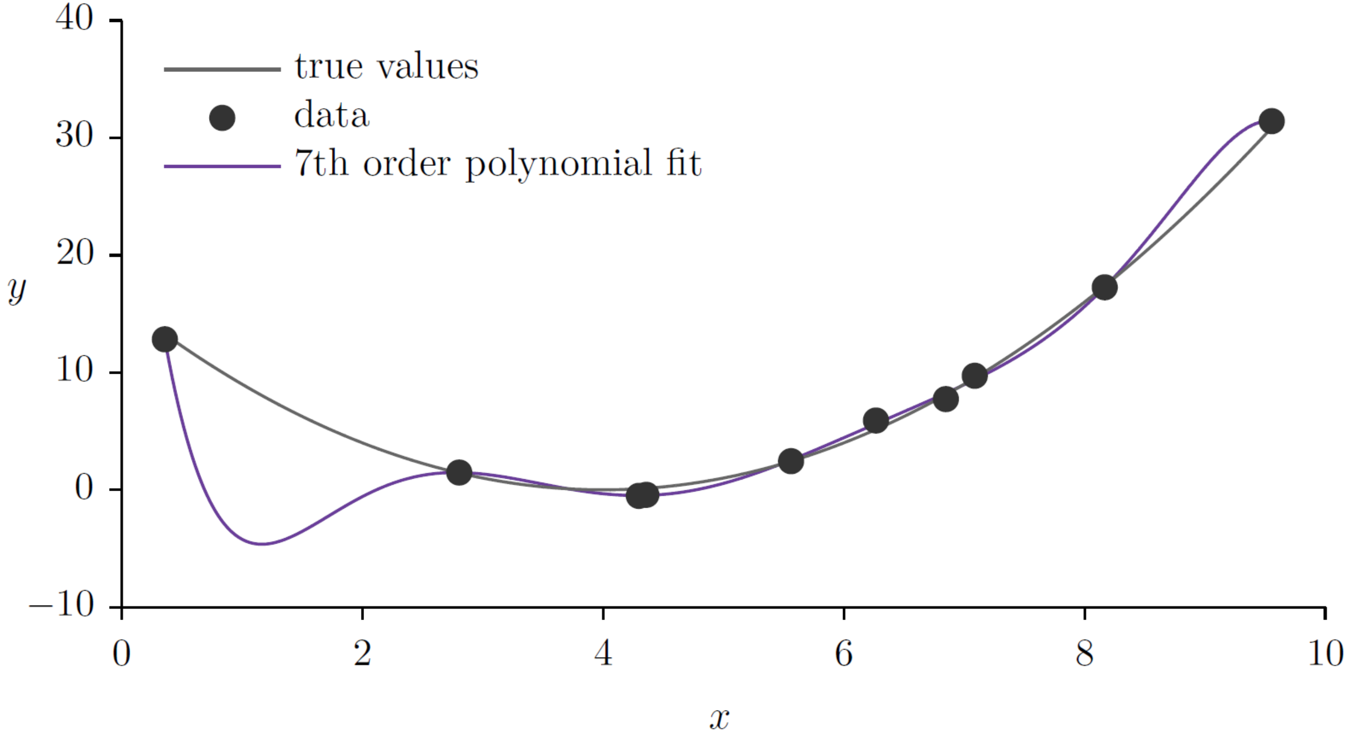
   Digits taken from a CD-ROM published by George Marsaglia.

A random number:

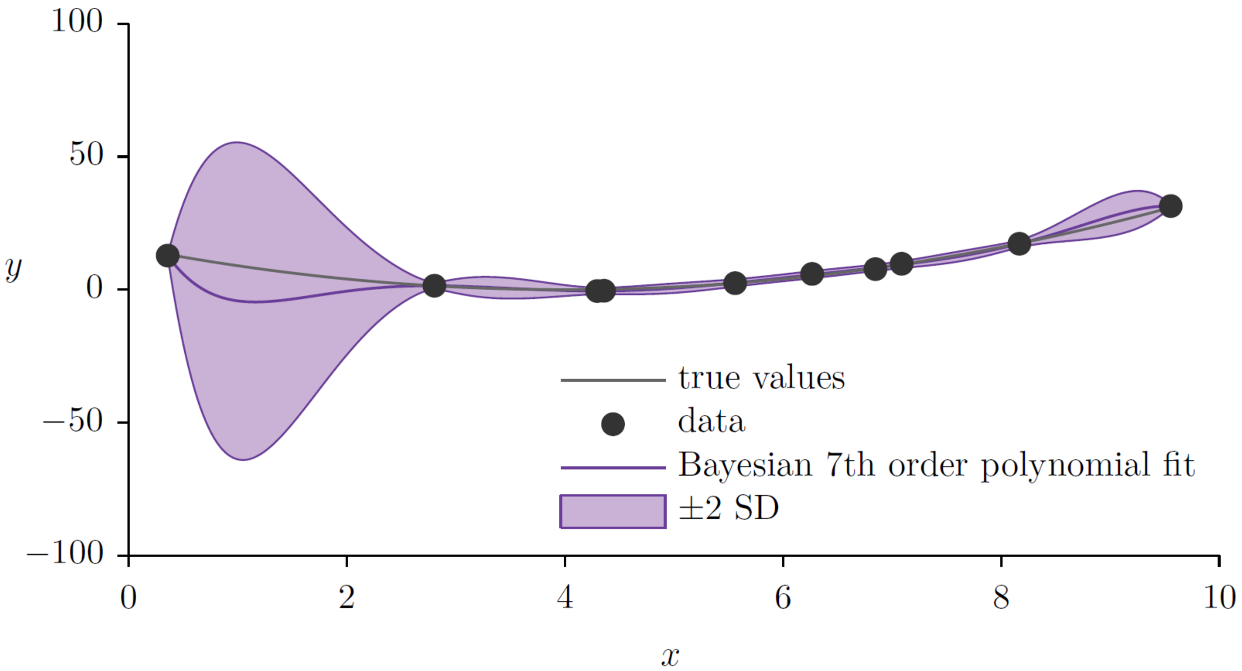
1. is epistemic (of course, computation is always conditional on prior knowledge);
2. is useful to foil a malicious adversary (of which there are few in numerics); and
3. is never the minimiser of an expected loss.

Integration beats optimisation

The naïve fitting of models to data performed by optimisation can lead to overfitting.



Bayesian averaging over ensembles of models reduces overfitting, and provides more honest estimates of uncertainty.

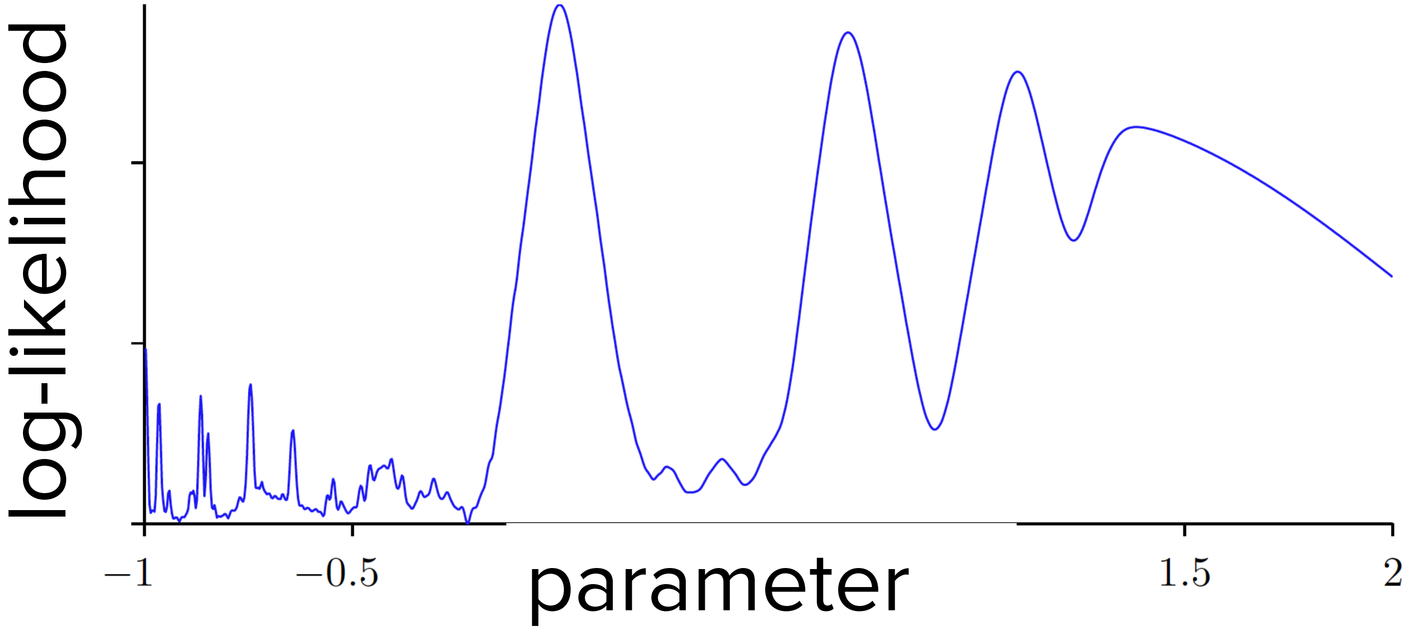


With parameters, our model is $p(f_\star, \mathcal{D}, \theta)$. Then

$$p(f_\star \mid \mathcal{D}) = \frac{p(f_\star, \mathcal{D})}{p(\mathcal{D})} = \frac{\int p(f_\star, \mathcal{D}, \theta)\,\mathrm{d}\theta}{p(\mathcal{D})} = \frac{\int p(f_\star \mid \mathcal{D}, \theta)\, p(\mathcal{D} \mid \theta)\, p(\theta)\,\mathrm{d}\theta}{p(\mathcal{D})}$$

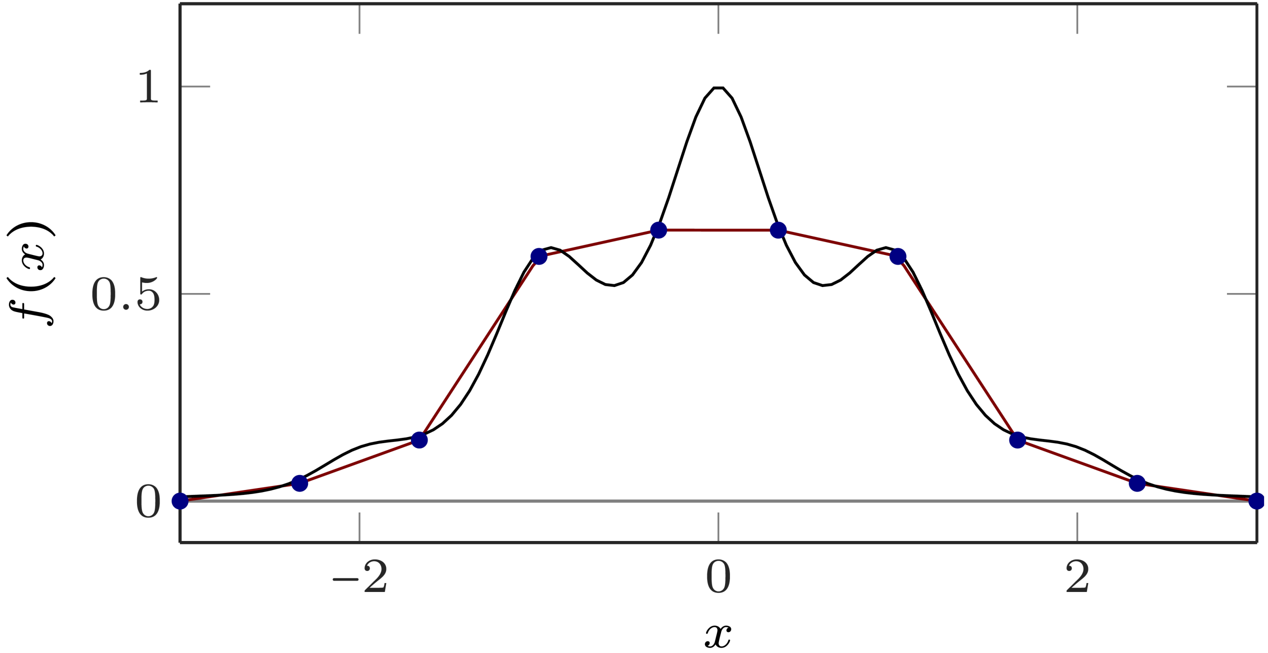
1. $p(f_\star \mid \mathcal{D})$ is called the posterior for $f_\star$; this is our goal.
2. $p(f_\star \mid \mathcal{D}, \theta)$ are the predictions given θ.
3. $p(\theta)$ is called the prior for θ.
4. $p(\mathcal{D} \mid \theta)$ is called the likelihood of θ.
5. $p(\mathcal{D}) = \int p(\mathcal{D} \mid \theta)\, p(\theta)\,\mathrm{d}\theta$ is called the evidence, or marginal likelihood.

Averaging requires integrating over the many possible states of the world consistent with data: this is often non-analytic.

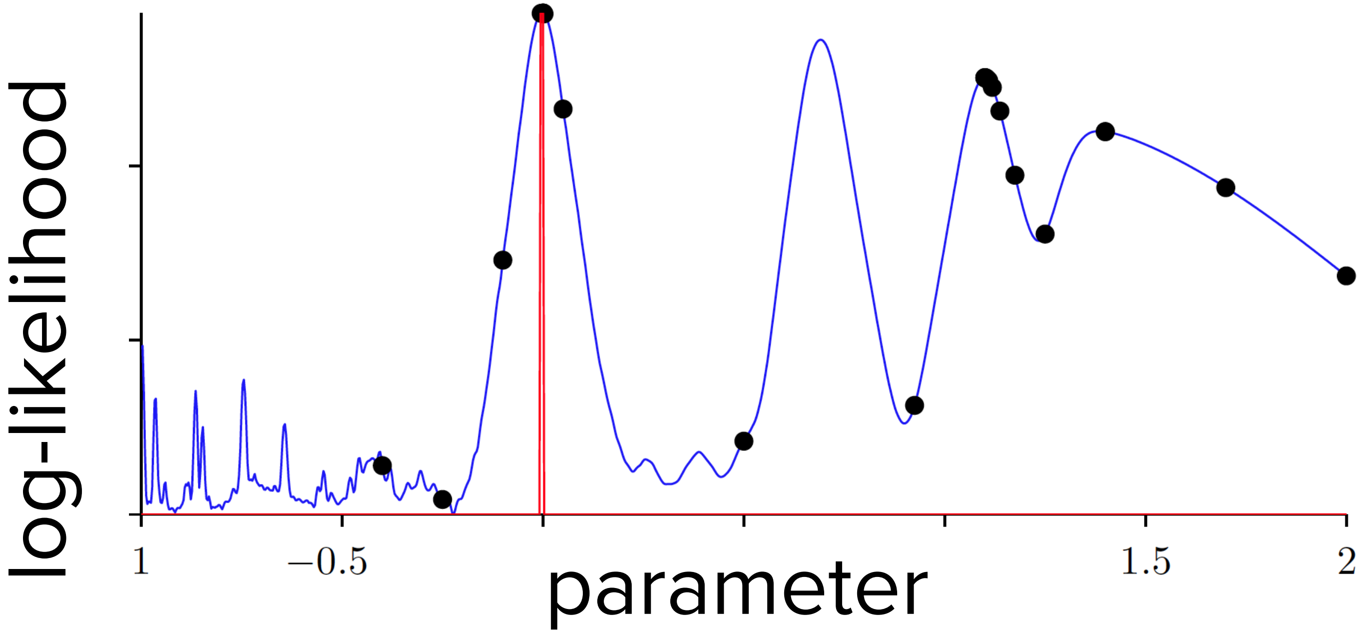


Numerical integration (quadrature) is ubiquitous.

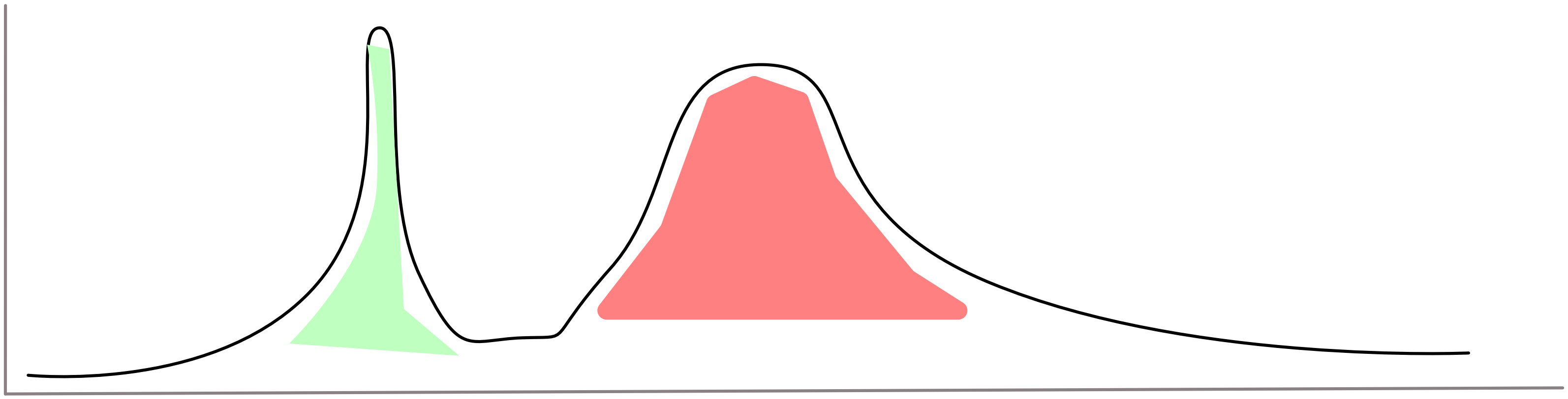
$$f(x) := \exp\left(-(\sin(3x))^2 - x^2\right)$$



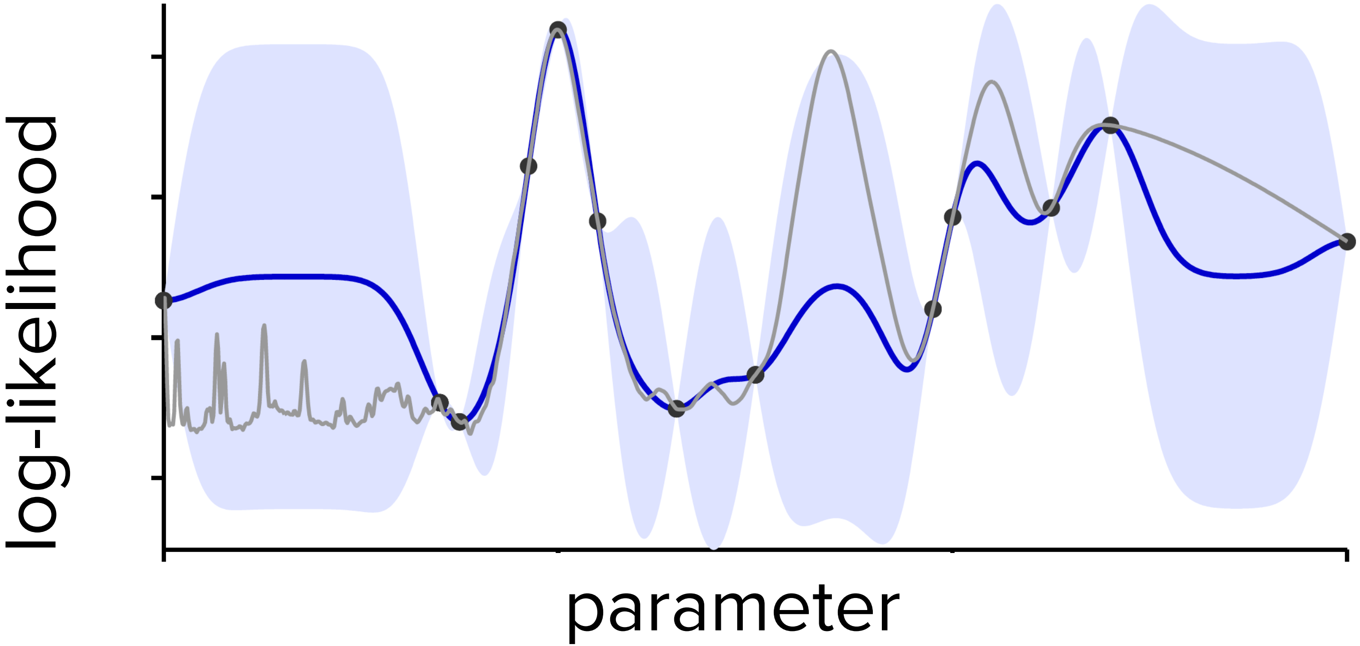
Optimisation is an unreasonable way of estimating a multi-modal or broad likelihood integrand.



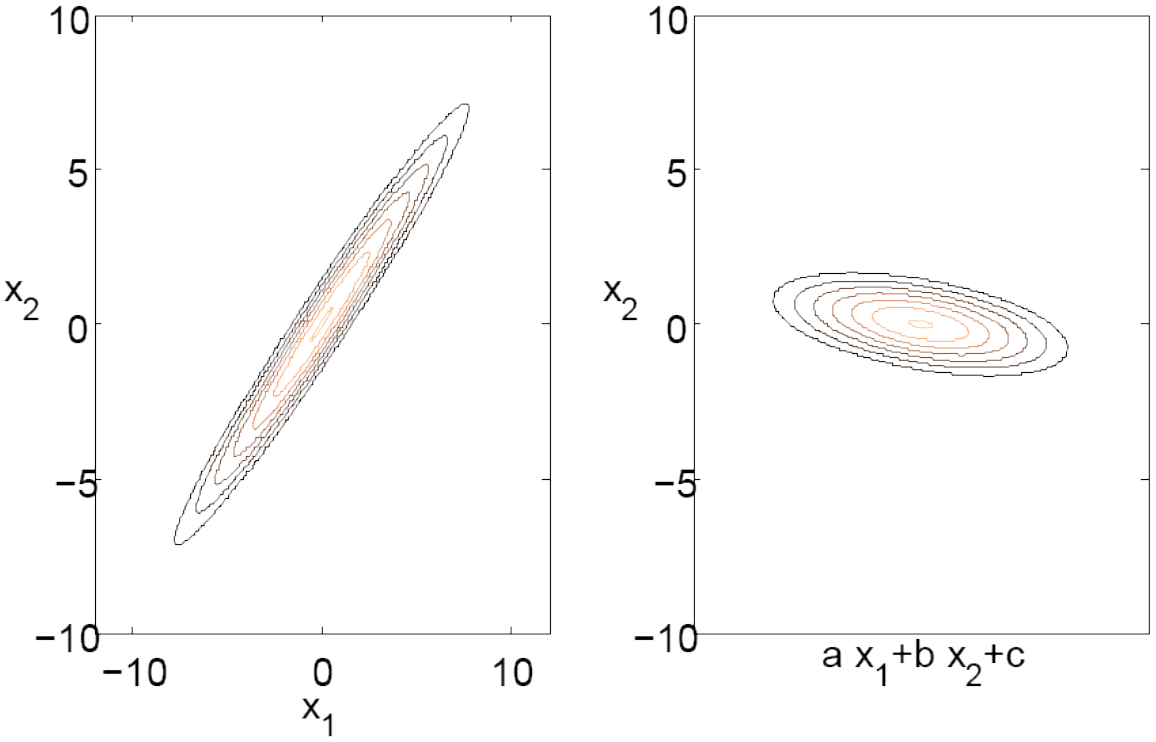
If optimising, flat optima are often a better representation of the integral than narrow optima.



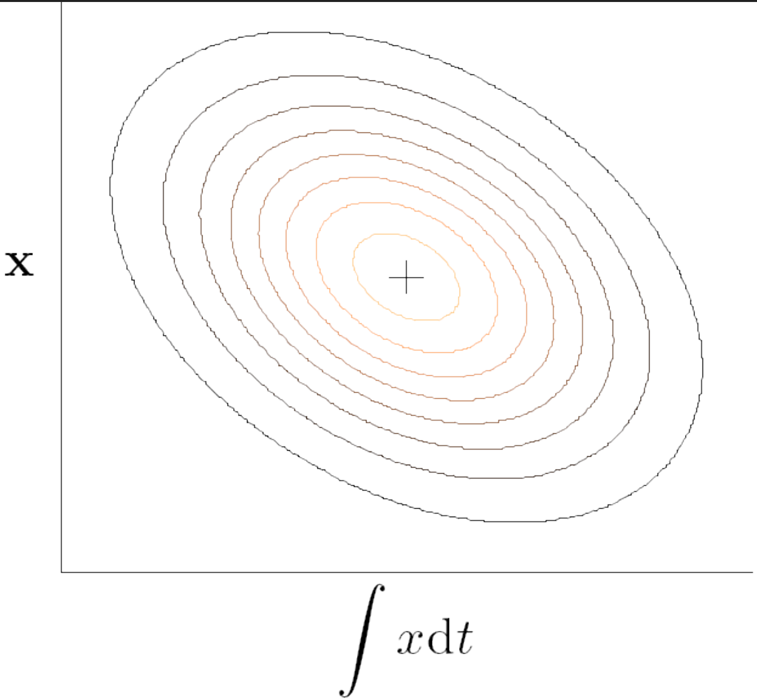
Bayesian quadrature makes use of a Gaussian process surrogate for the integrand (the same as you might use for Bayesian optimisation).



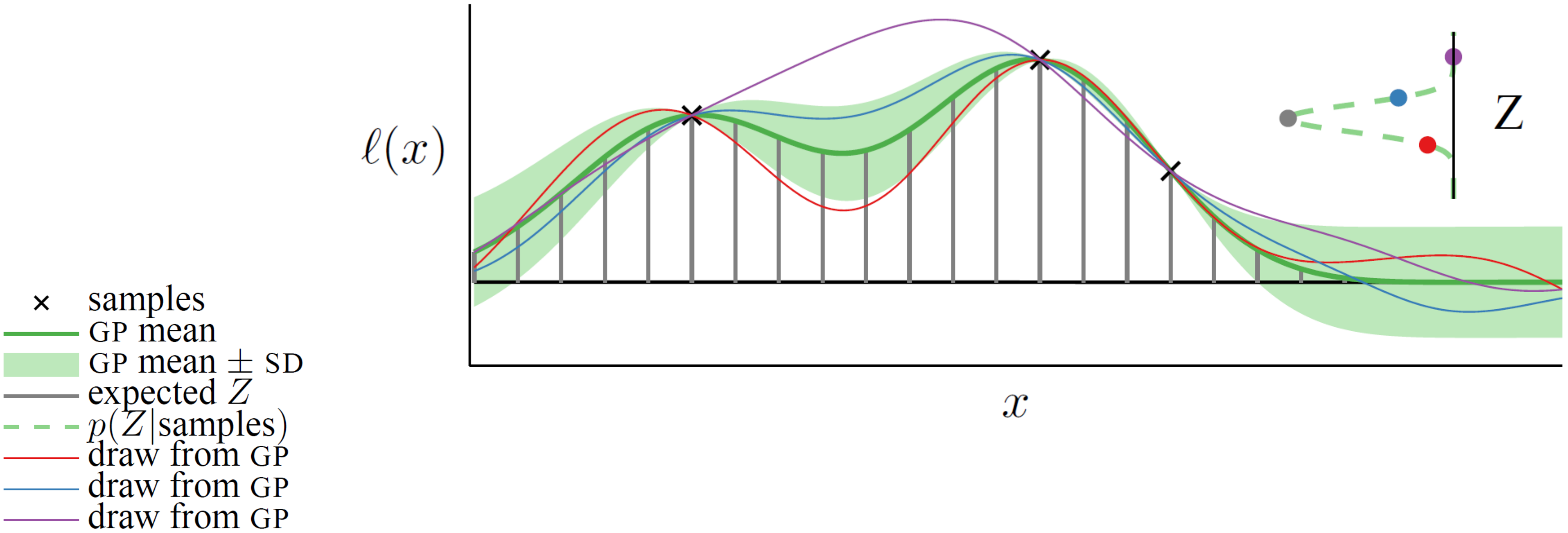
Gaussian distributed variables are joint Gaussian with any affine transform of them.



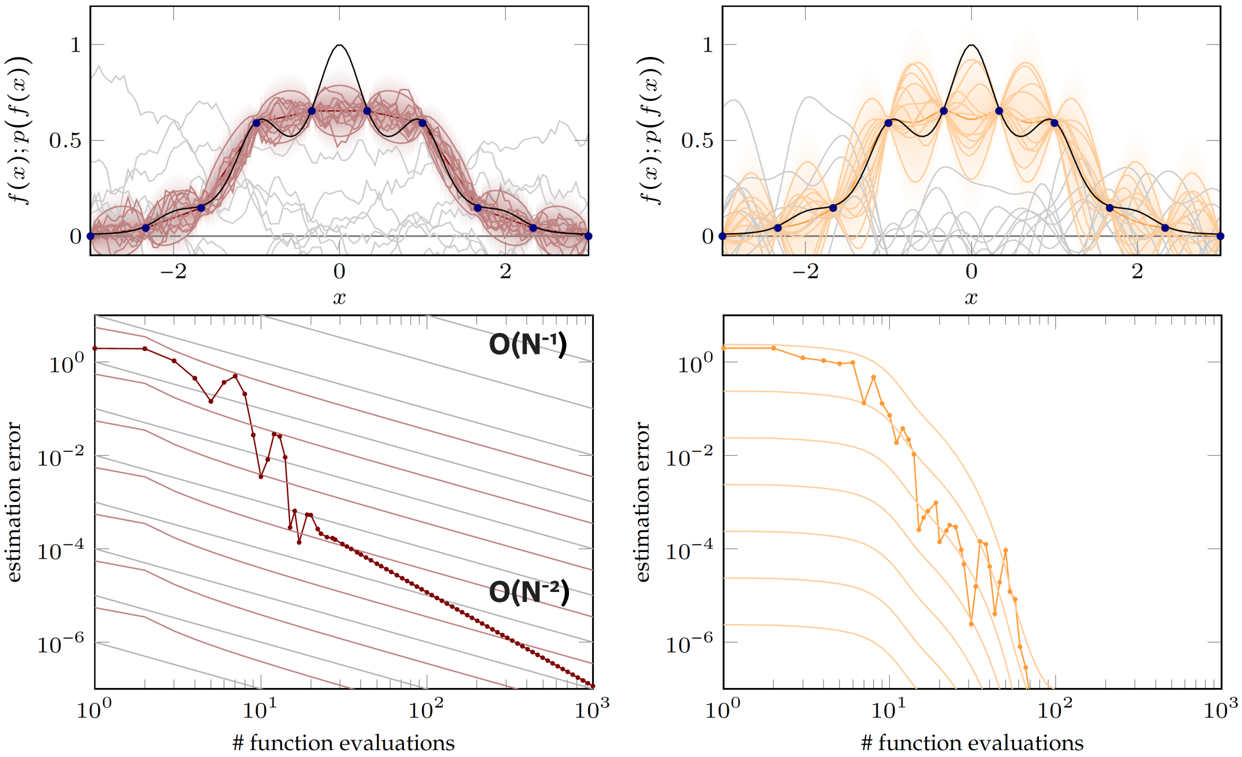
A function over which we have a Gaussian process is joint Gaussian with any integral or derivative of it, as integration and differentiation are linear.



We can use observations of an integrand ℓ in order to perform inference for its integral, Z: this is known as Bayesian Quadrature.



Bayesian quadrature generalises and improves upon traditional quadrature.



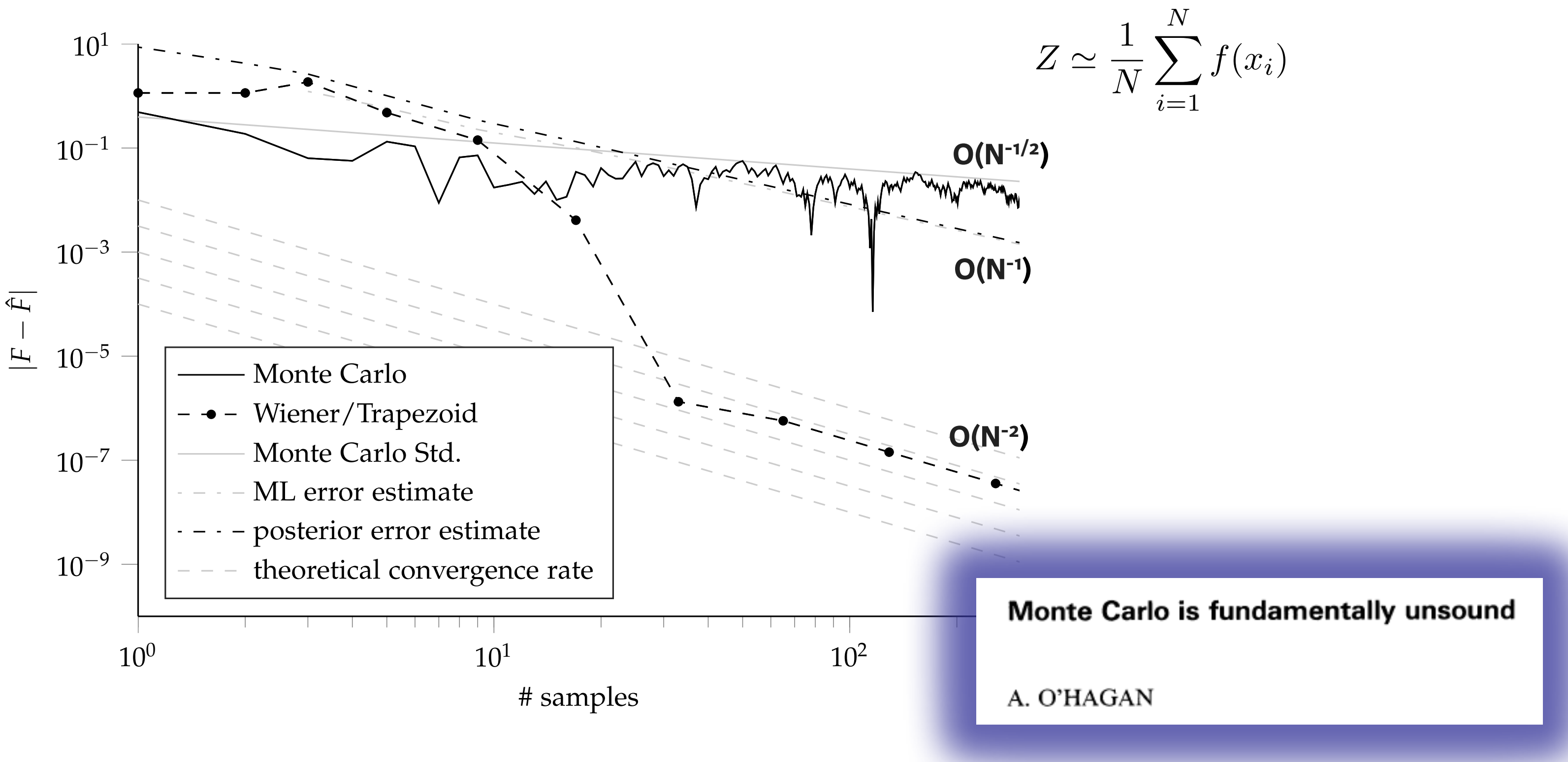
Quiz: what is the convergence rate of Monte Carlo?

1. $O(\exp(-N))$
2. $O(\exp(-N^{-1/2}))$
3. $O(N^{-1})$
4. $O(N^{-1/2})$

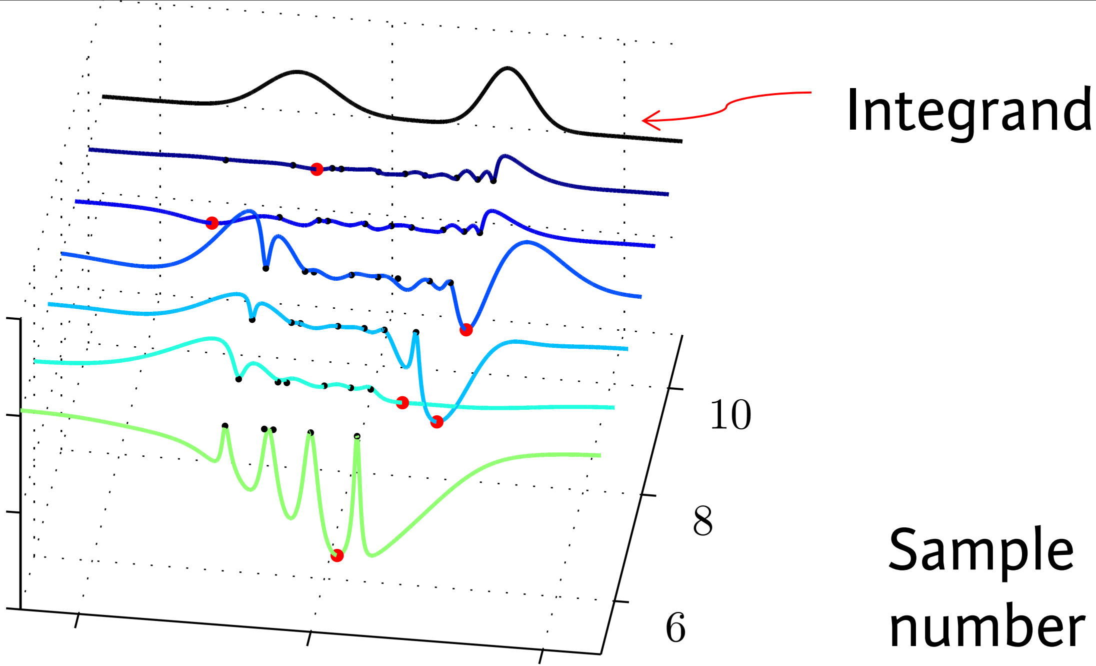
Quiz: what is the convergence rate of Monte Carlo?

1. $O(\exp(-N))$
2. $O(\exp(-N^{-1/2}))$
3. $O(N^{-1})$
4. $O(N^{-1/2})$

The trapezoid rule ($O(N^{-2})$) has empirically better scaling than Monte Carlo ($O(N^{-½})$).

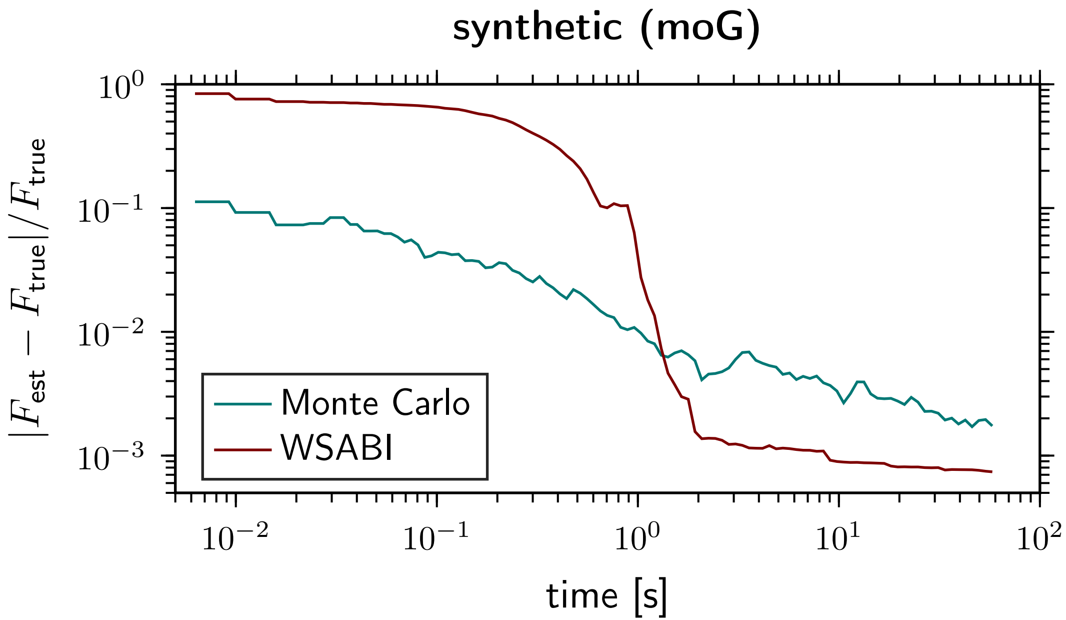


Probabilistic numerics views the selection of samples as a decision problem.



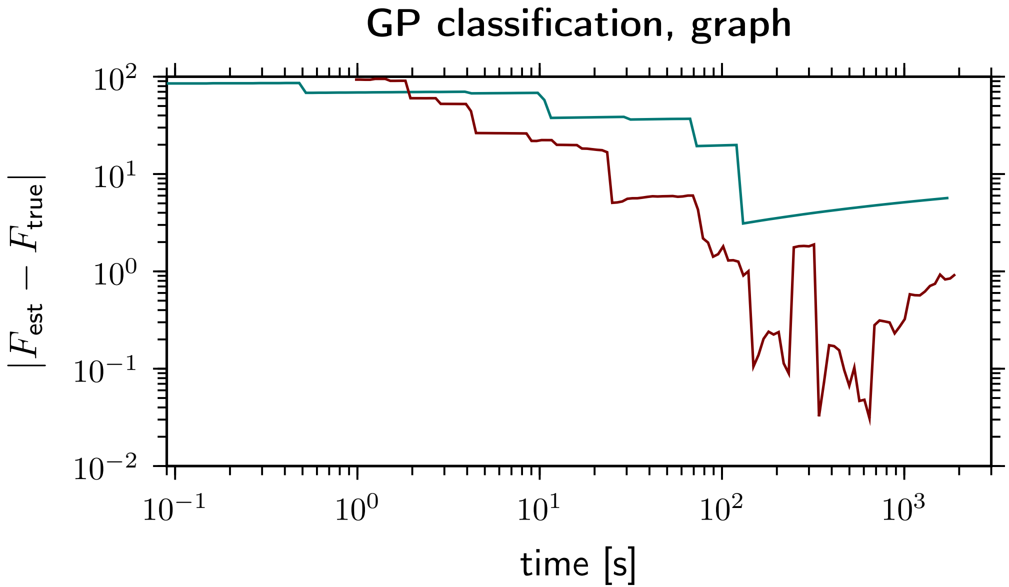
Osborne, M. A., Duvenaud, D. K., Garnett, R., Rasmussen, C. E., Roberts, S. J., & Ghahramani, Z. (2012). Active learning of model evidence using Bayesian quadrature. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 46–54).

Our method (Warped Sequential Active Bayesian Integration) converges quickly in wall-clock time for a synthetic integrand.



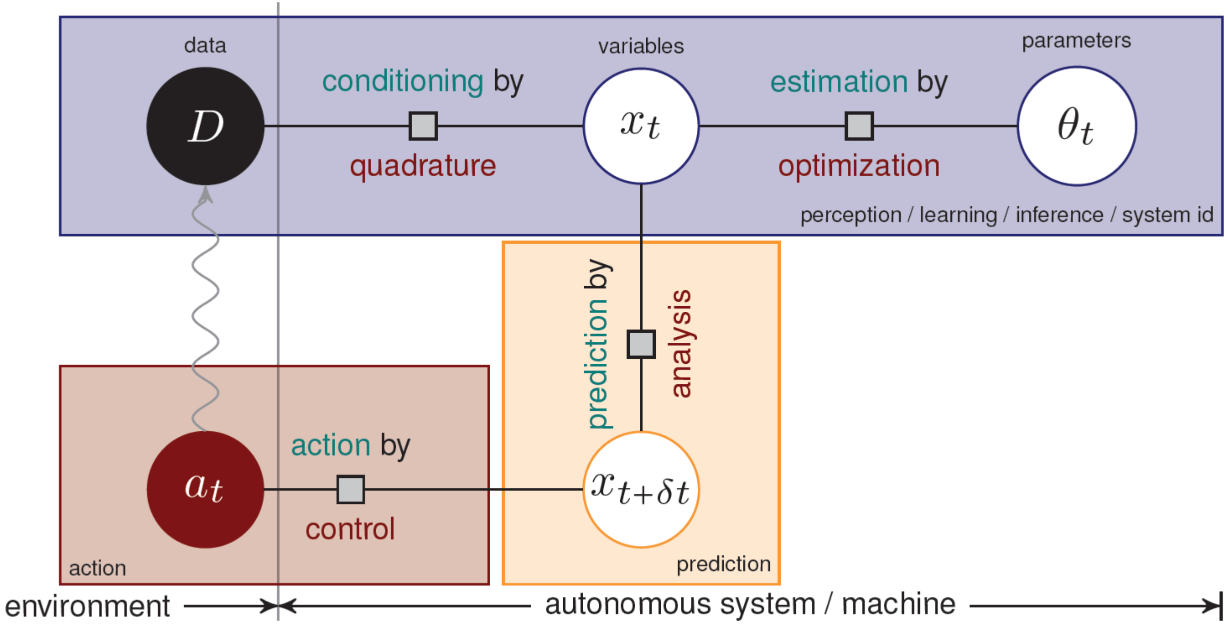
Gunter, T., Osborne, M. A., Garnett, R., Hennig, P., & Roberts, S. J. (2014). Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. In Advances in Neural Information Processing Systems (NIPS).

WSABI-L converges quickly in integrating out hyperparameters in a Gaussian process classification problem (CiteSeer^x data).



Gunter, T., Osborne, M. A., Garnett, R., Hennig, P., & Roberts, S. J. (2014). Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature. In Advances in Neural Information Processing Systems (NIPS).

Probabilistic numerics offers the propagation of uncertainty through numerical pipelines.



Probabilistic numerics treats computation as a decision.