Machine Learning and the Physical World

Lecture 7 : Probabilistic Nummerics

Carl Henrik Ek - che29@cam.ac.uk

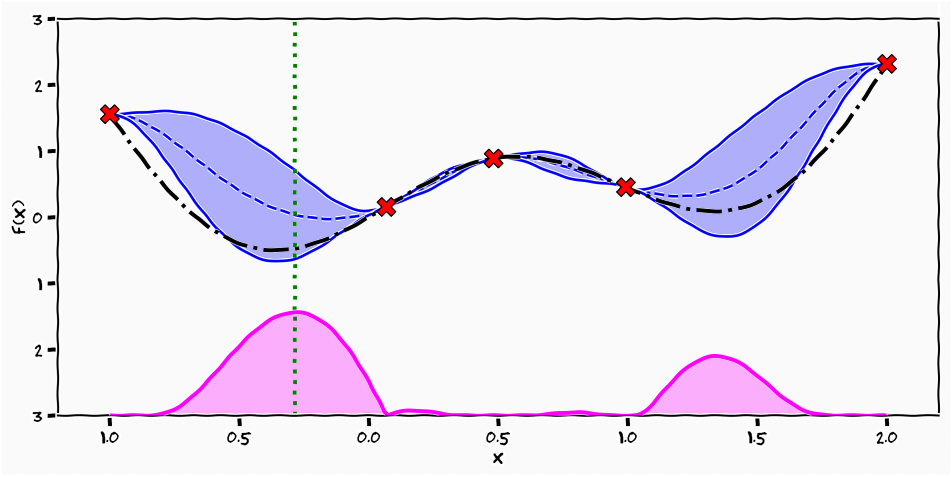
29th of October, 2021

http://carlhenrik.com

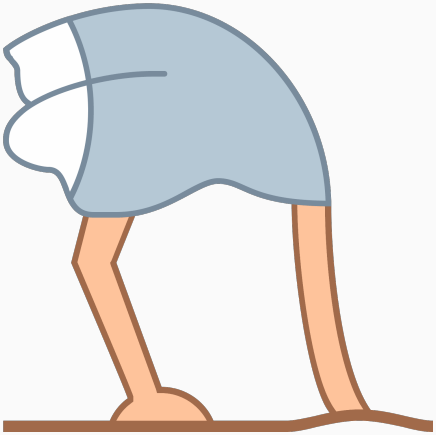
The role of Uncertainty/Ignorance

$$p(y) = \int p(y \mid f) p(f) \mathrm{d}f$$

Bayesian Optimisation



What do we do with uncertainty?



Today

2021-10-29

The Marginal Likelihood

Semantics

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{\int p(y \mid \theta)p(\theta)\mathrm{d}\theta}$$

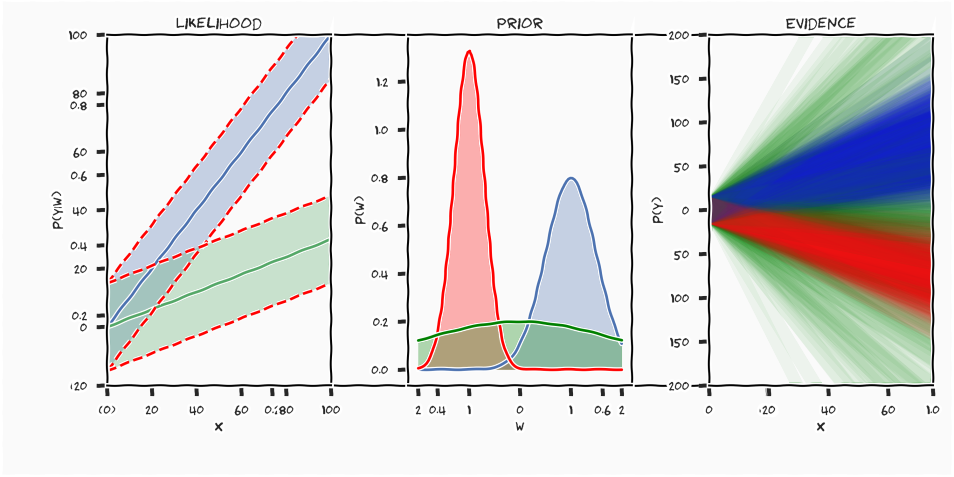
Likelihood How much evidence is there in the data for a specific hypothesis

Prior What are my beliefs about different hypothesis

Posterior What is my updated belief after having seen data

Evidence What is my belief about the data

Regression Model



Marginalisation

Next time you want to give your friends a compliment, tell them that you have completely marginalised them from your life

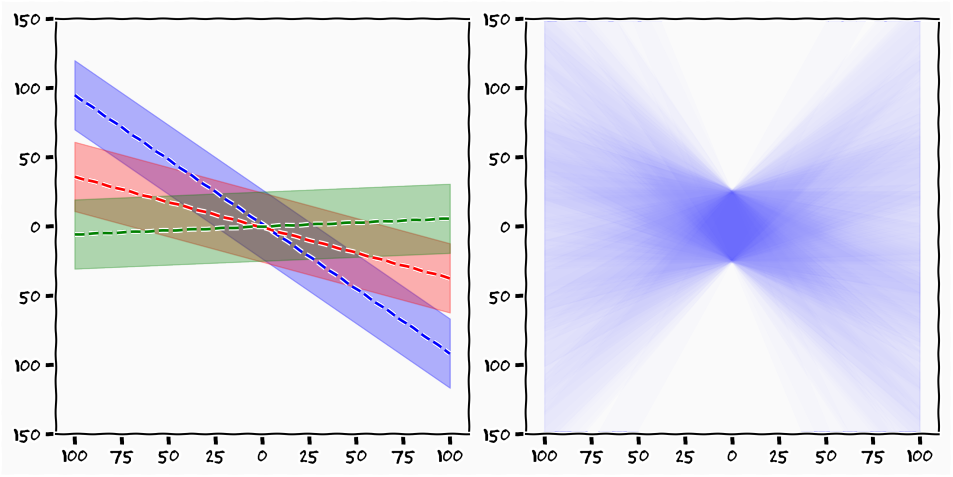
Linear Linear Model

$$p(y_i|x_i, \mathbf{w}) = \mathcal{N}(w_0 + w_1 \cdot x_i, \beta^{-1})$$

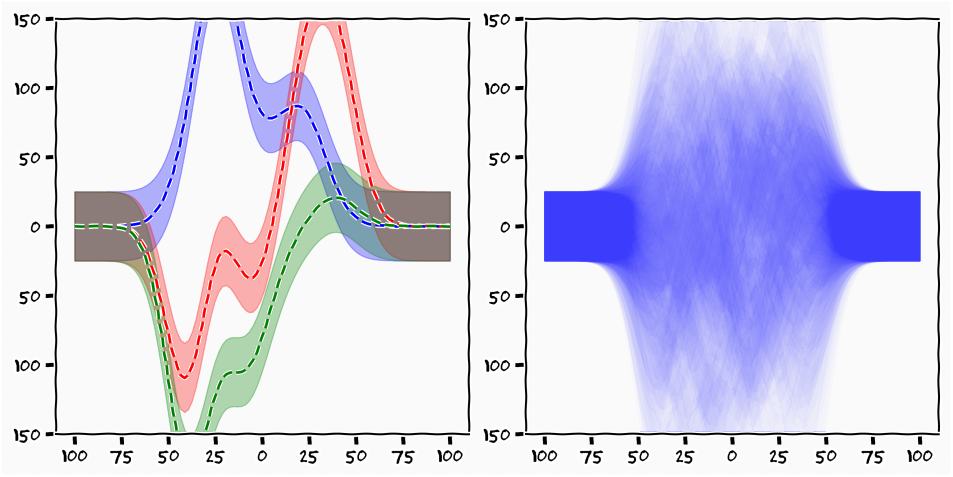
Basis function

$$p(y_i|x_i, \mathbf{w}) = \mathcal{N}(\sum_{i=1}^{6} w_i \phi(x_i), \beta^{-1})$$

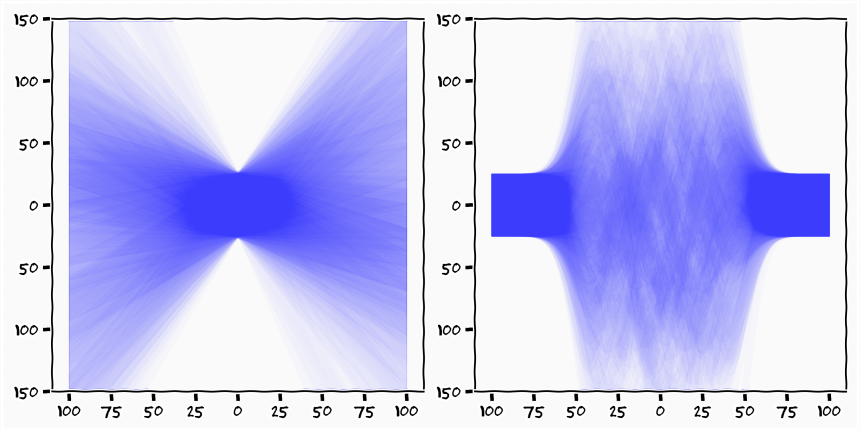
Linear Linear Regression



Linear Regression



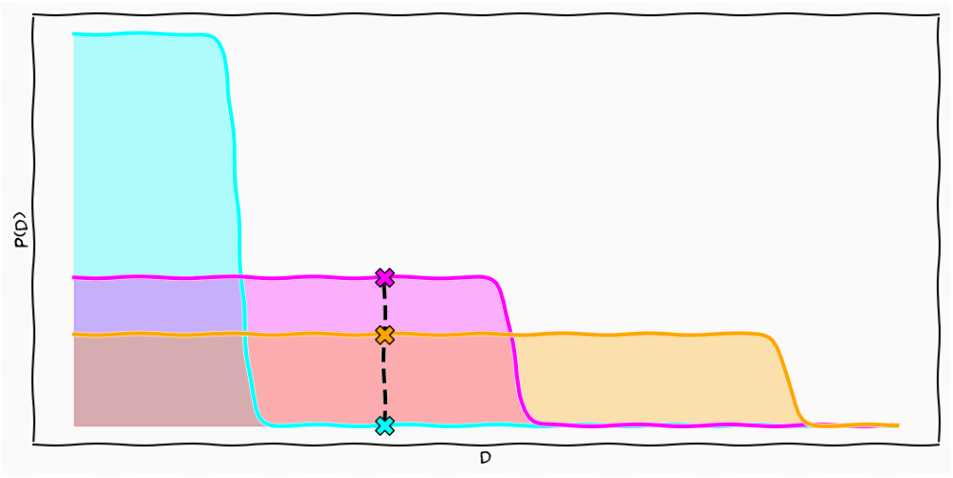
Evidence



$$p(\mathcal{Y}) = \int p(\mathcal{Y}|\mathbf{W})p(\mathbf{W})\mathrm{d}\mathbf{W}$$

Probabilities are a zero-sum game

The MacKay Plot Mackay, 1991



Occams Razor

Occams Razor

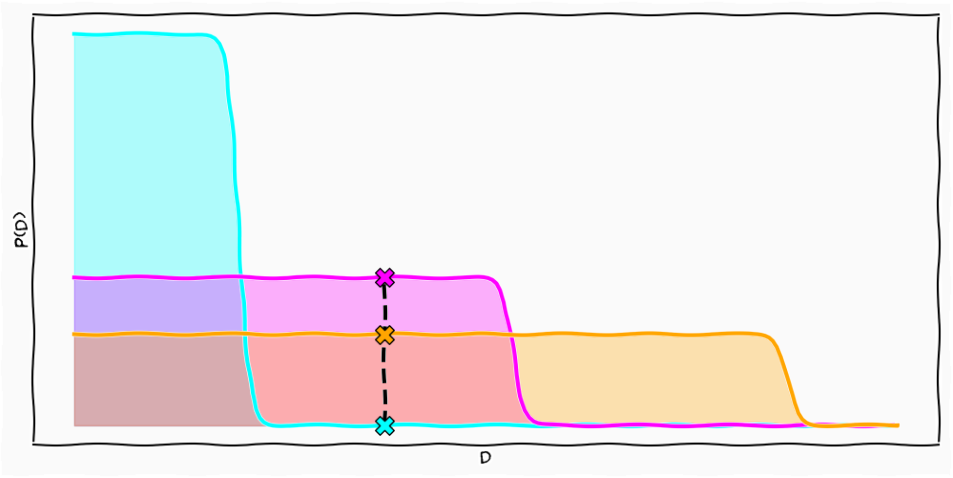
Definition (Occams Razor)
"All things being equal, the simplest solution tends to be the best one"

– William of Ockham

What is Simple?[1]

[1] https://www.imdb.com/title/tt8132700/

The MacKay Plot Mackay, 1991



Unsupervised Learning [Lawrence, 2005]

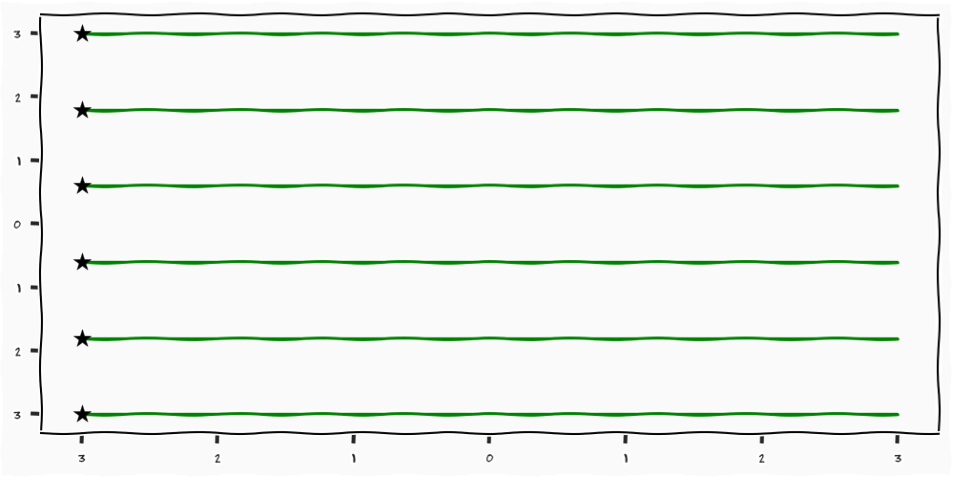
- Regression

$$p(y \mid x) = \int p(y \mid f)p(f \mid x)\mathrm{d}f$$

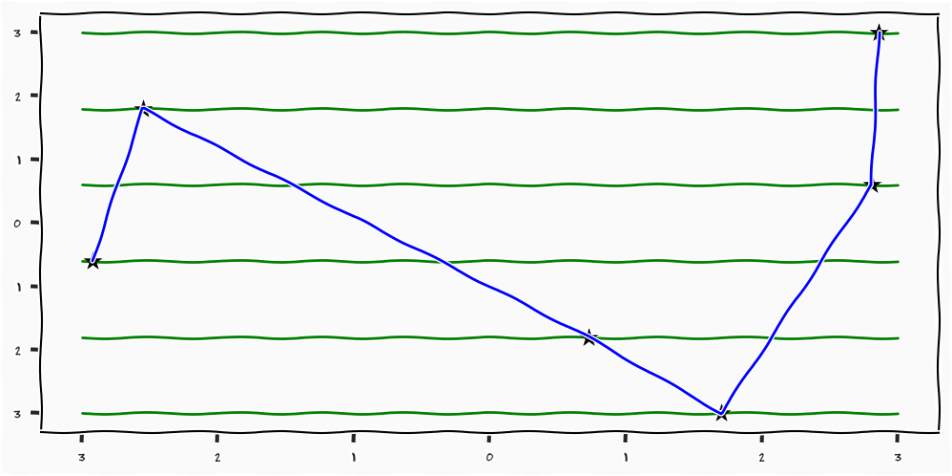
- "Unsupervised" Learning

$$p(y) = \int p(y \mid f)p(f \mid x)p(x)\mathrm{d}f\mathrm{d}x$$

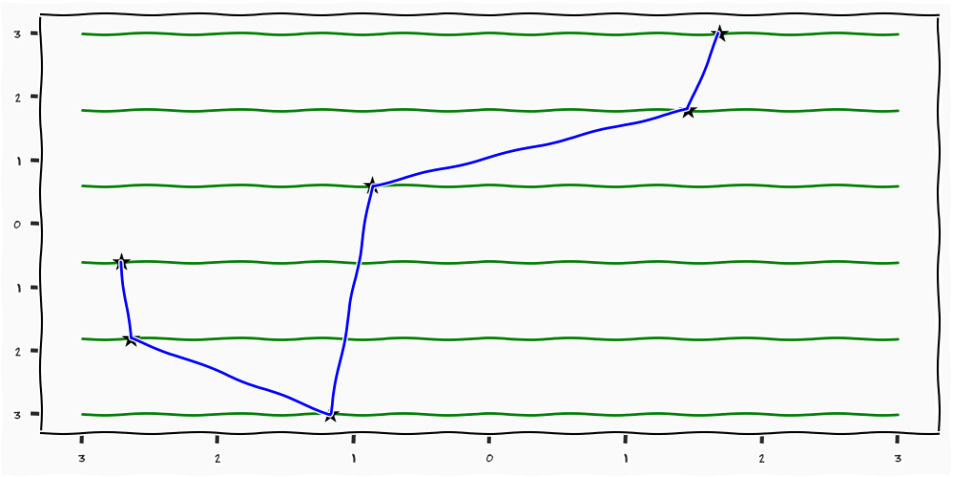
Unsupervised Learning



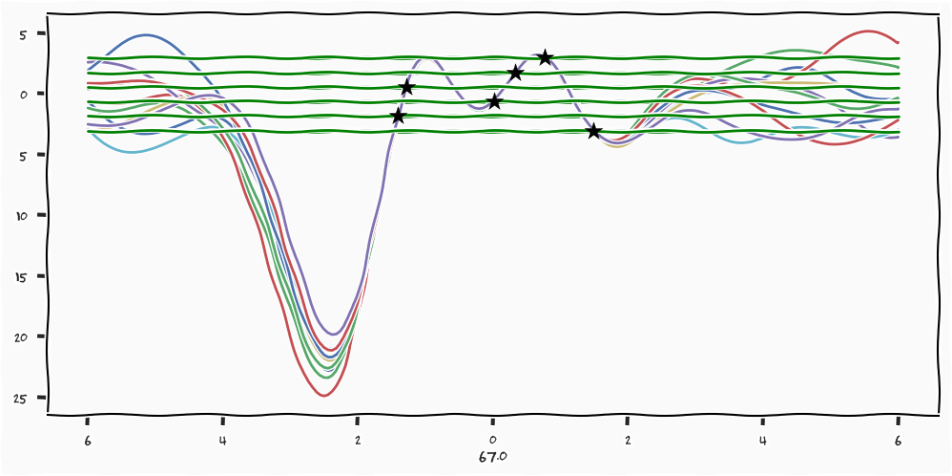
Unsupervised Learning



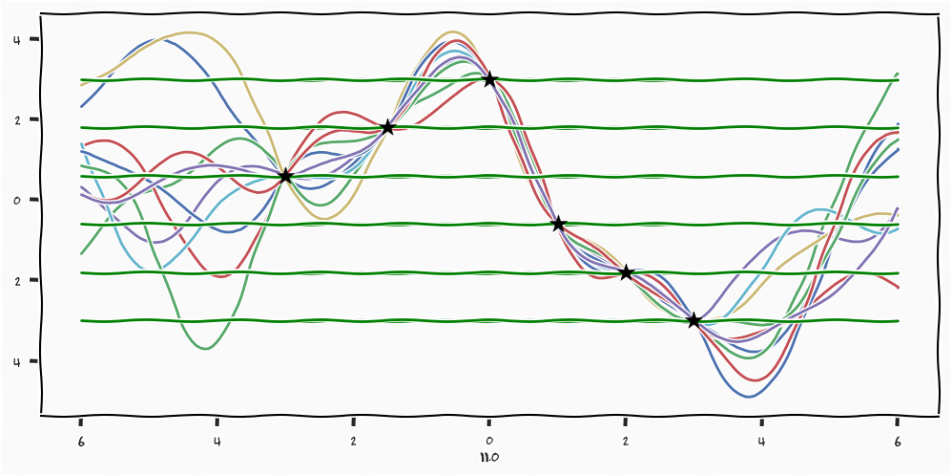
Unsupervised Learning



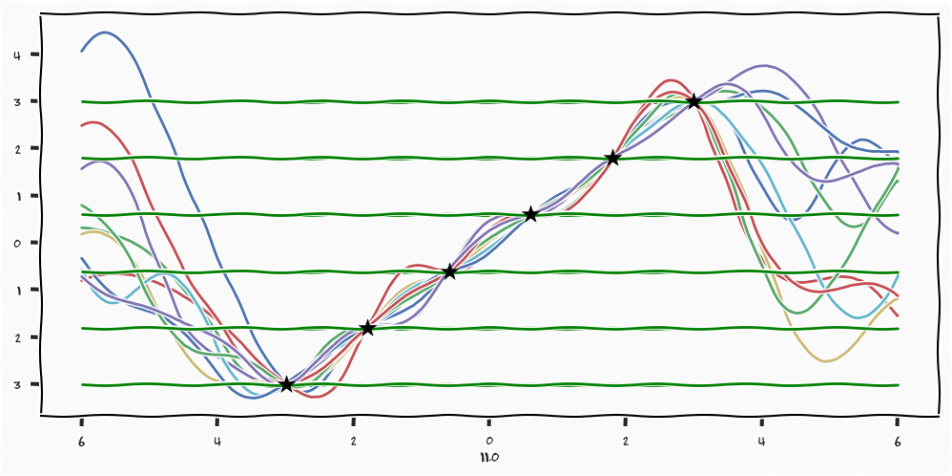
Gaussian Process Latent Variable Model [Lawrence, 2005]



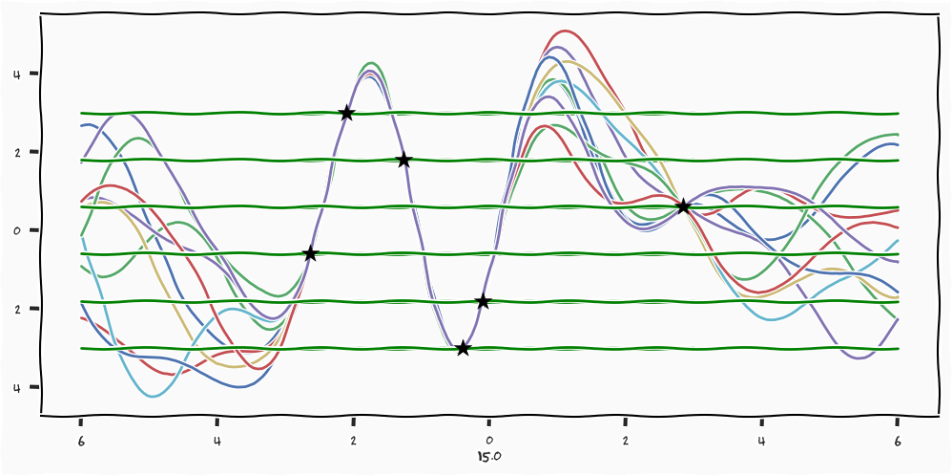
Gaussian Process Latent Variable Model [Lawrence, 2005]



Gaussian Process Latent Variable Model [Lawrence, 2005]



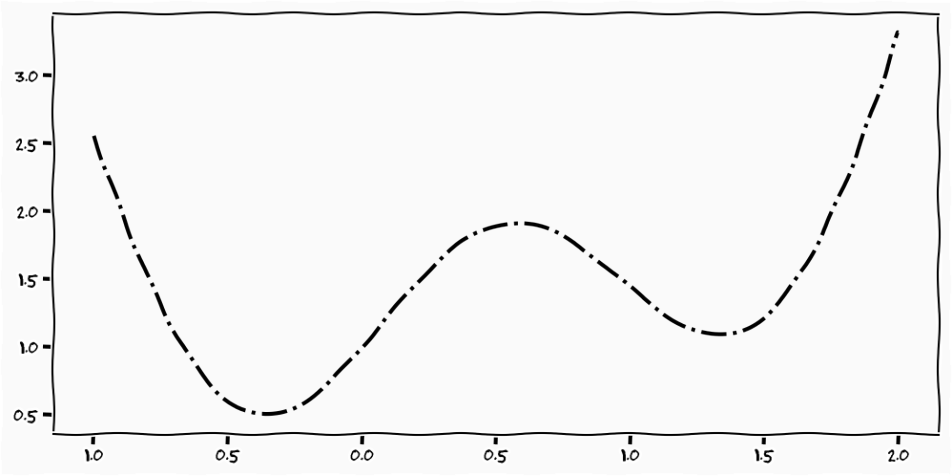
Gaussian Process Latent Variable Model [Lawrence, 2005]



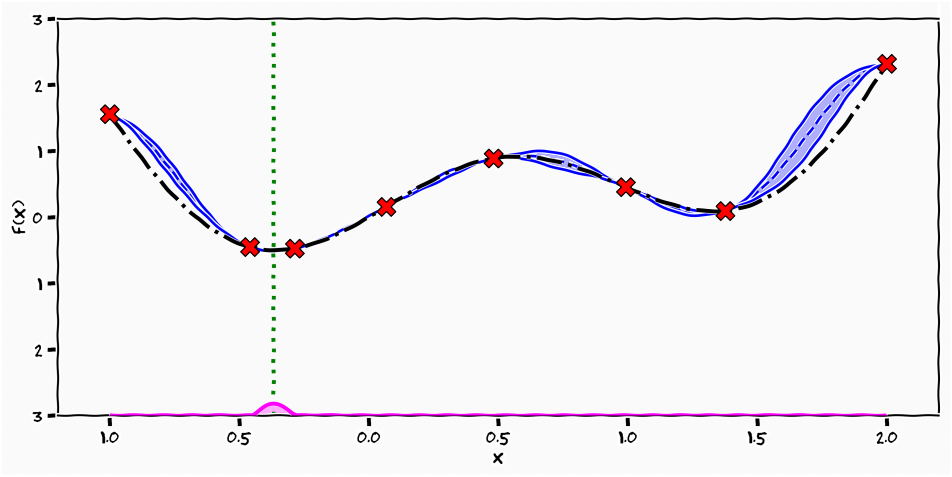
Today

I believe in . . .

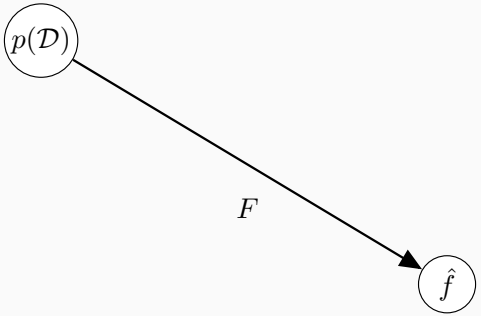
Functions



Optimisation

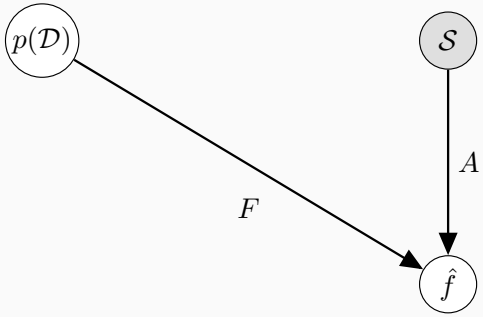


Formalisation [Cockayne et al., 2017]



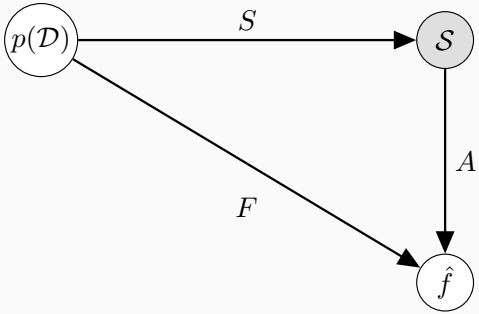
$$F : p(\mathcal{D}) \to p(\mathcal{Y}|\mathcal{X})$$

Formalisation [Cockayne et al., 2017]



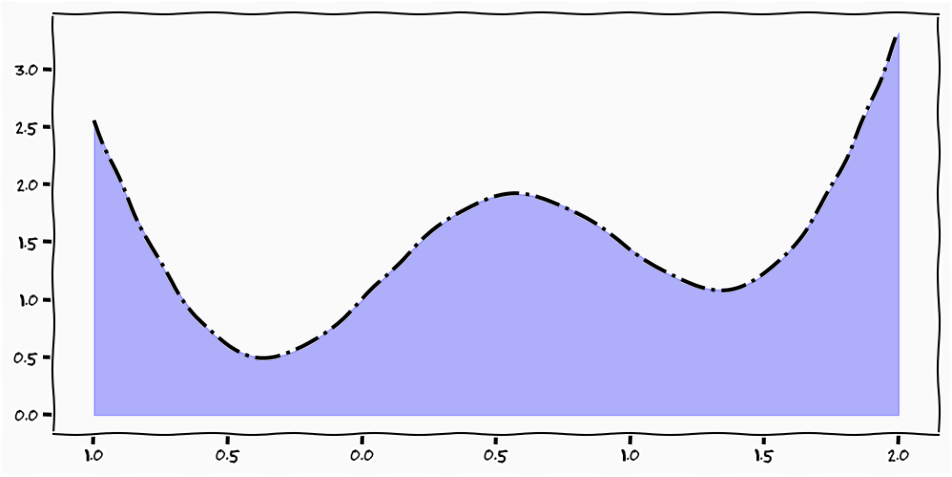
$$A \circ \mathcal{S} \approx F \circ p(\mathcal{D})$$

Formalisation [Cockayne et al., 2017]

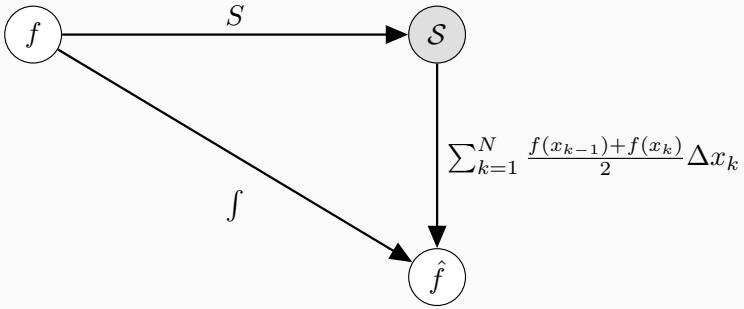


$$A \circ S \circ p(\mathcal{D}) \approx F \circ p(\mathcal{D})$$

Function Quantity

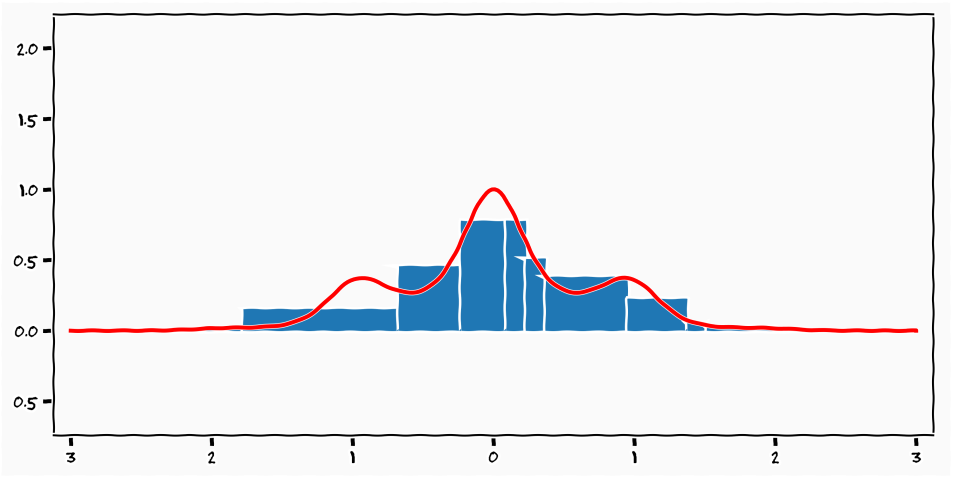


Quadrature [Cockayne et al., 2017]

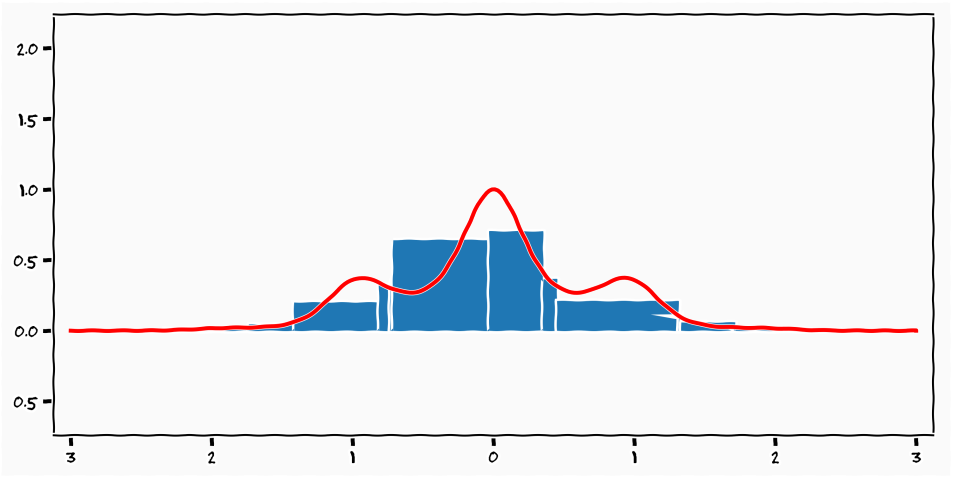


$$A \circ \mathcal{S} \approx \int f(x) \mathrm{d}x$$

Quadrature



Quadrature



Numerical Computation

A numerical method estimates a function's latent property given the result of computations.

A numerical method estimates a function's latent property given the result of computations.

$\frac{\text{Numerical algorithms}}{\text{Statistical inference}}$ takes data in the form of $\frac{\text{evaluations of computations}}{\text{measurements of observed variables}}$ and aims to return predictions of the quantity of interest.

A numerical method estimates a function's latent property given the result of computations.

$\frac{\text{Numerical algorithms}}{\text{Statistical inference}}$ takes data in the form of $\frac{\text{evaluations of computations}}{\text{measurements of observed variables}}$ and aims to return predictions of the quantity of interest.

Should we think about computation as inference?

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Decision when to stop computation

Why Probabilistic Numerics?

"[round-off errors] are strictly very complicated but uniquely defined number theoretical functions [of the inputs], yet our ignorance of their true nature is such that we best treat them as random variables."
– Neumann et al., 1947

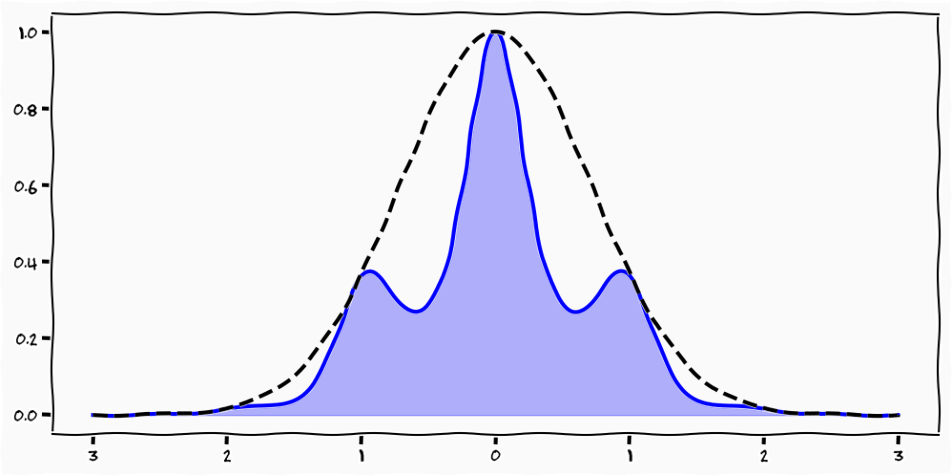
Probabilistic Numerics

Quantity of interest

$$F := \int_{-3}^{3} \underbrace{e^{-(sin(3x))^2 - x^2}}_{f(x)} \mathrm{d}x$$

- $f(x)$ fully specified and deterministic
- F is deterministic
- F cannot be computed analytically

Integration

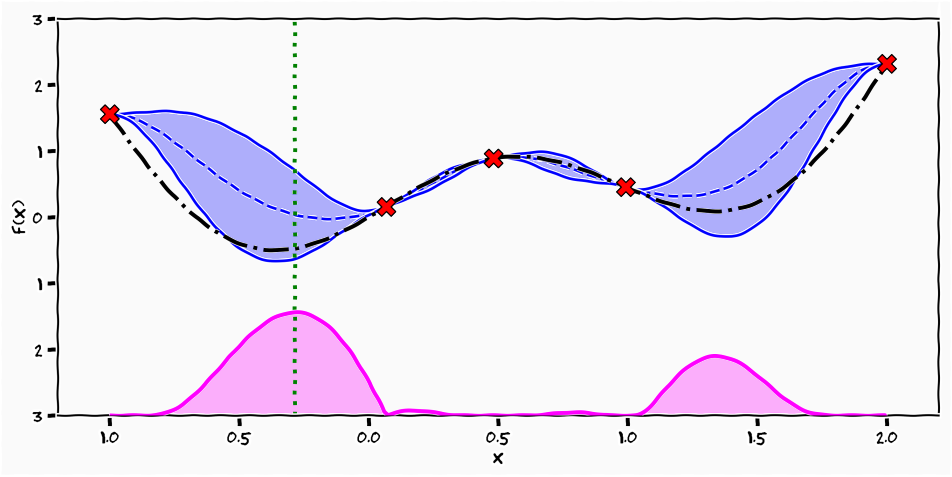


What we would like

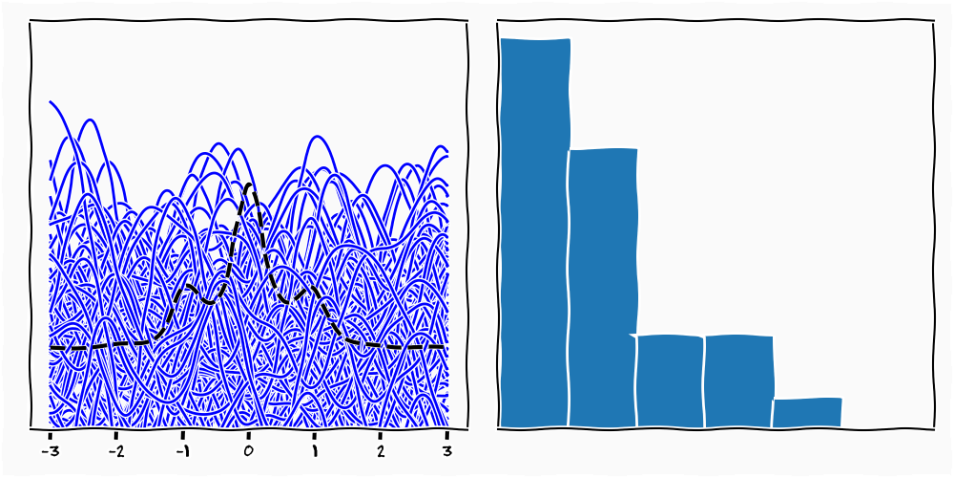
$$p(F \mid Y)$$

- given that I have seen data Y what is my belief about the integral
- allows for "active learning"
- exploration/exploitation etc.

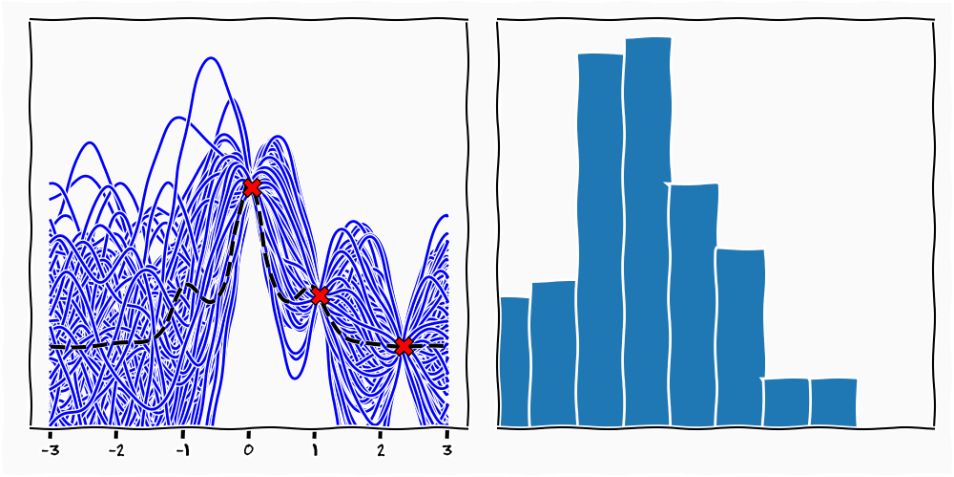
Emulation



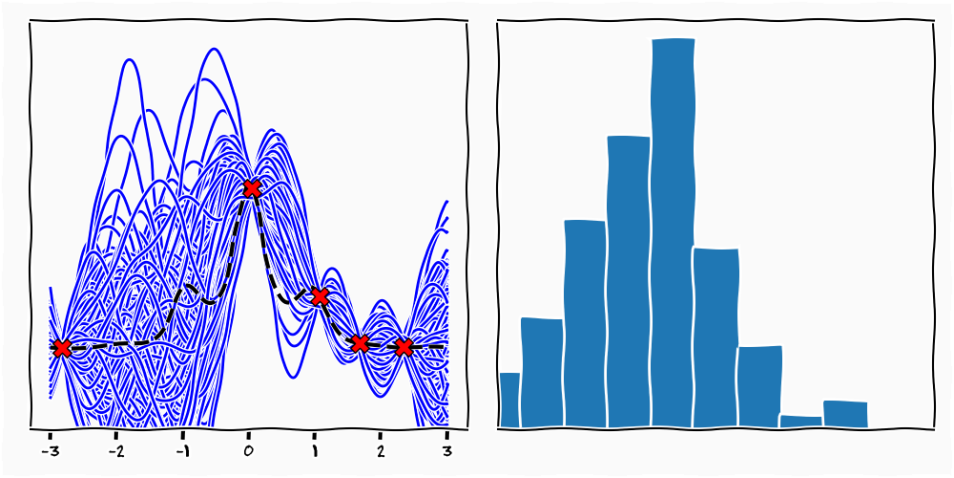
Quadrature



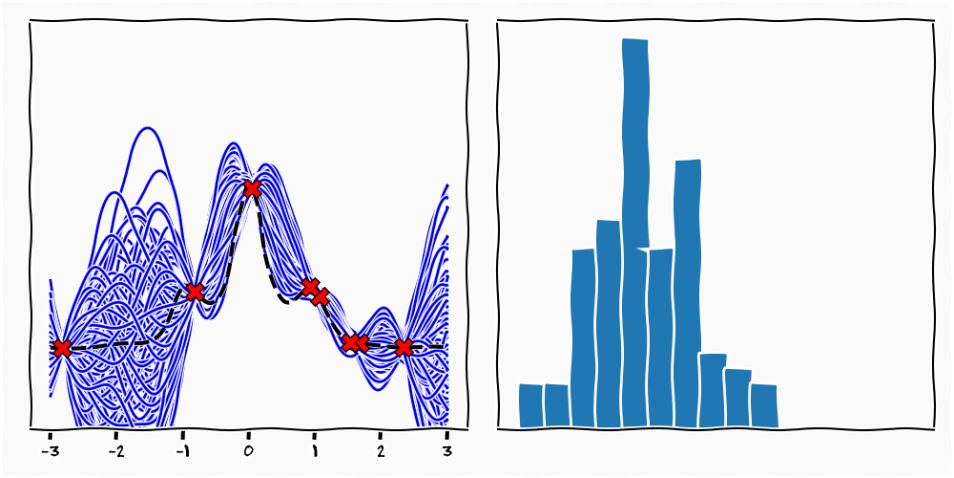
Quadrature



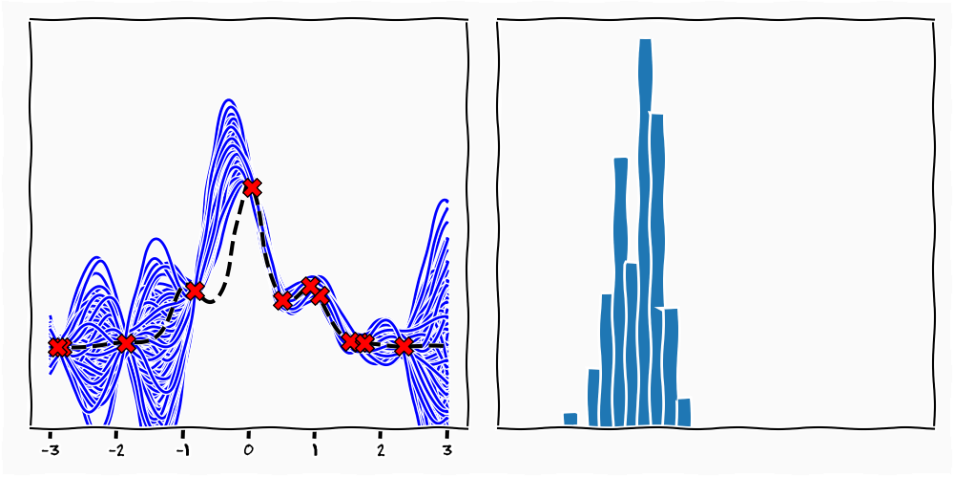
Quadrature



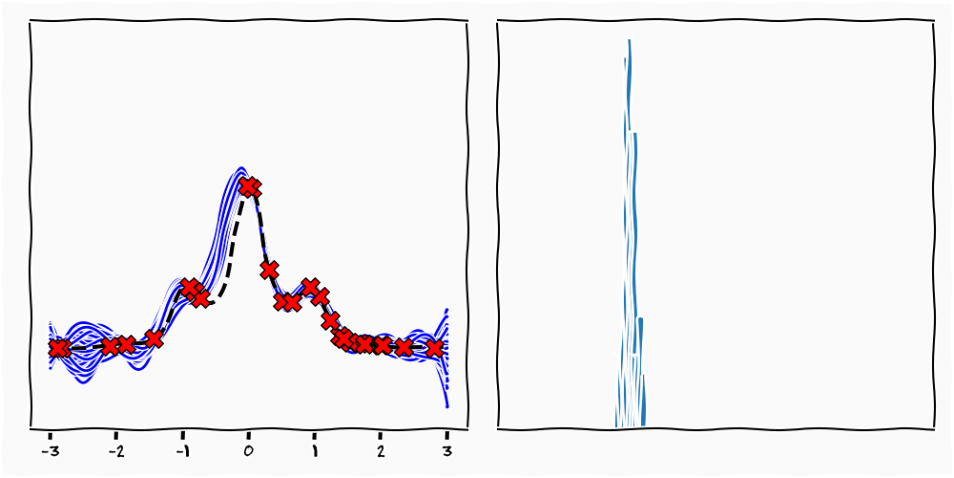
Quadrature



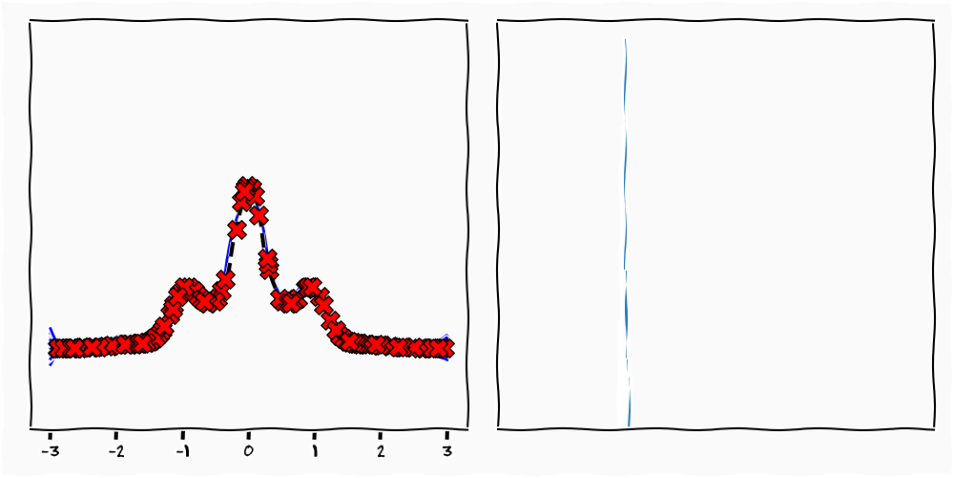
Quadrature



Quadrature



Quadrature



Bayesian Quadrature [O'Hagan, 1991]

$$p(F, Y) = \int p(F \mid f) p(Y \mid f) p(f) \mathrm{d}f$$

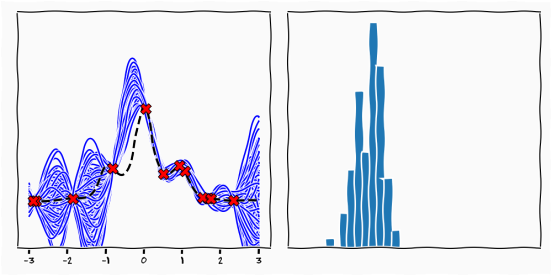
Bayesian Quadrature [O'Hagan, 1991]

$$
\begin{aligned}
p(F, Y) &= \int p(F \mid f) p(Y \mid f) p(f) \mathrm{d}f \\
&= \int \delta \left(F - \int_{\mathcal{X}} f \mathrm{d}x \right) \prod_{i}^{N} \delta(y_i - f(x_i)) p(f) \mathrm{d}f
\end{aligned}
$$

$$p(F, Y) = \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_X \\ \int m_X(x)\mathrm{d}x \end{bmatrix}, \begin{bmatrix} k(X,X) & \int k(X,x)\mathrm{d}x \\ \int k(x,X)\mathrm{d}x & \int\int k(x,x')\mathrm{d}x\mathrm{d}x' \end{bmatrix}\right)$$

- We can derive $p(F \mid Y)$ through our normal conditioning procedure
- $p(F \mid Y) = \mathcal{N}(\mu_F, k_F)$ is a uni-variate Gaussian

Information Operator[2]

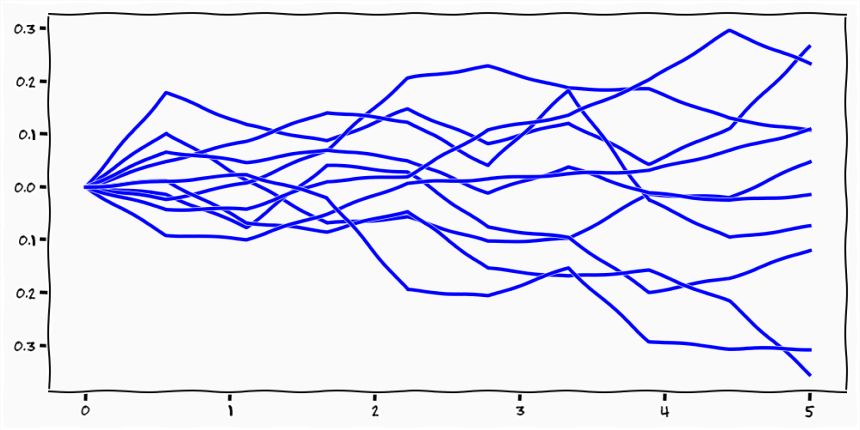


Integrand variance $\alpha(x) = k(x, x)$

Integral Variance Reduction $\alpha(x) = k_F(X, X) - k_F(X, x)$

[2]sometimes called a "Design Rule"

Choice of Covariance



$$p(f) = \mathcal{GP}(\mathbf{0}, \theta^2(\min(x, x') - \kappa)$$

Quadrature Rule

$$\mathbb{E}[F] = \mathbb{E}_{p(f|Y}\left[\int f(x)\mathrm{d}x\right] = \sum_{i=1}^{N-1} \frac{x_{i+1} - x_i}{2}(f(x_{i+1}) + f(x_i))$$

Quadrature Rule

$$\mathbb{E}[F] = \mathbb{E}_{p(f|Y}\left[\int f(x)\mathrm{d}x\right] = \sum_{i=1}^{N-1} \frac{x_{i+1} - x_i}{2}(f(x_{i+1}) + f(x_i))$$

- This is the normal trapezoid rule!!!

Quadrature Rule

$$\mathbb{E}[F] = \mathbb{E}_{p(f|Y}\left[\int f(x)\mathrm{d}x\right] = \sum_{i=1}^{N-1} \frac{x_{i+1} - x_i}{2}(f(x_{i+1}) + f(x_i))$$

- This is the normal trapezoid rule!!!
- The algorithm is now tied to the function!!!!

Quadrature Rule

$$\mathbb{E}[F] = \mathbb{E}_{p(f|Y}\left[\int f(x)\mathrm{d}x\right] = \sum_{i=1}^{N-1} \frac{x_{i+1} - x_i}{2}(f(x_{i+1}) + f(x_i))$$

- This is the normal trapezoid rule!!!
- The algorithm is now tied to the function!!!!
- We can do inference over where to sample!!!!!!!!

Code

```
def minimize(fun, x0, args=(), method=None,
             jac=None, hess=None,
             hessp=None, bounds=None,
             constraints=(), tol=None,
             callback=None, options=None):
```

method Nelder-Mead, Powell, CG, BFGS, Newton-CG, L-BFGS-B, TNC , COBYLA , SLSQP , trust-constr , dogleg , trust-ncg , trust-exact , trust-krylov

Probabilistic Numerics [Hennig et al., 2015]

- There are tons of numerical algorithms for every problem under the sun

Probabilistic Numerics [Hennig et al., 2015]

- There are tons of numerical algorithms for every problem under the sun
- They work really well

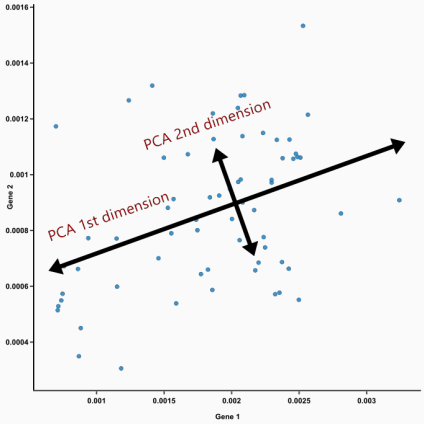
Probabilistic Numerics [Hennig et al., 2015]

- There are tons of numerical algorithms for every problem under the sun
- They work really well
- They give different results on the same problem

Probabilistic Numerics [Hennig et al., 2015]

- There are tons of numerical algorithms for every problem under the sun
- They work really well
- They give different results on the same problem
- *what is the prior they implement?*

PCA


PCA 2nd dimension
PCA 1st dimension
Gene 2
Gene 1
0.0016
0.0014
0.0012
0.001
0.0008
0.0006
0.0004
0.001
0.0015
0.002
0.0025
0.003

Summary

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth"

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth"
- Computation is Machine Learning

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth"
- Computation is Machine Learning
- Why?

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth"
- Computation is Machine Learning
- Why?
  - efficiency

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth"
- Computation is Machine Learning
- Why?
  - efficiency
  - down-stream tasks, uncertainty in computation should be part of decision

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth"
- Computation is Machine Learning
- Why?
  - efficiency
  - down-stream tasks, uncertainty in computation should be part of decision
  - learning/understanding algorithms in relation to problems/data

Numerical Computations[3]

Quadrature given $f(x_i)$ estimate $\int_a^b f(x)\mathrm{d}x$

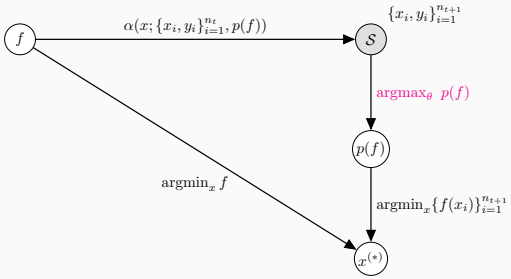
Linear Algebra given $As = y$ estimate x s.t. $Ax = b$

Optimisation given $\nabla f(x_i)$ estimate x s.t. $\nabla f(x) = 0$

Analysis given $f(x_i, t_i)$ estimate $x(t)$ s.t.

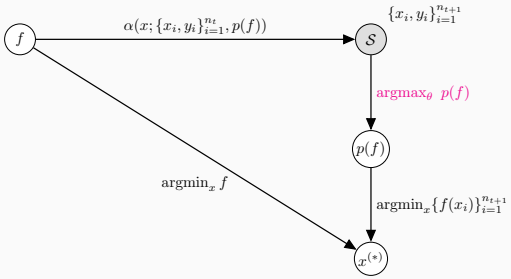
[3] https://www.cs.toronto.edu/~duvenaud/talks/odes_runge_kutta_nips.pdf

Is BO PN?



Yes it uses a probablistic model as a proxy for decision loop

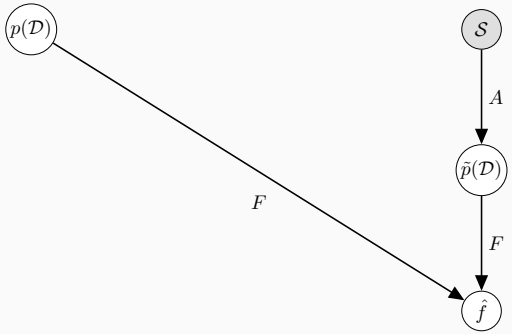
Is BO PN?



Yes it uses a probablistic model as a proxy for decision loop

No the probabilistic model is not over the quantity of interest

Formalisation



$$A \circ \mathcal{S} = \tilde{p}(\mathcal{D}) \approx p(\mathcal{D})$$

eof

References

- Cockayne, Jon, Chris Oates, Tim Sullivan, and Mark Girolami (2017). "Bayesian Probabilistic Numerical Methods". In: *CoRR*.
- Hennig, Philipp, Michael A Osborne, and Mark Girolami (July 2015). "Probabilistic numerics and uncertainty in computations". In: *Proc. R. Soc. A* 471.2179, p. 20150142.
- Lawrence, Neil D (2005). "Probabilistic non-linear principal component analysis with Gaussian process latent variable models". In: *Journal of Machine Learning Research* 6, pp. 1783–1816.

- Mackay, David J C (Dec. 1991). "Bayesian methods for adaptive models ". PhD thesis. California Institute of Technology: California Institute of Technology.
- Neumann, John von and H. H. Goldstine (1947). "Numerical Inverting of Matrices of High Order". In: *Bulletin of the American Mathematical Society* 53.11, pp. 1021–1100.
- O'Hagan, A. (Nov. 1991). "Bayes-Hermite quadrature". In: *Journal of Statistical Planning and Inference* 29.3, pp. 245–260.