UNIVERSITY OF CAMBRIDGE

Machine Learning and the Physical World

Lecture 7 : Probabilistic Numerics

Carl Henrik Ek - che29@cam.ac.uk

27th of October, 2022

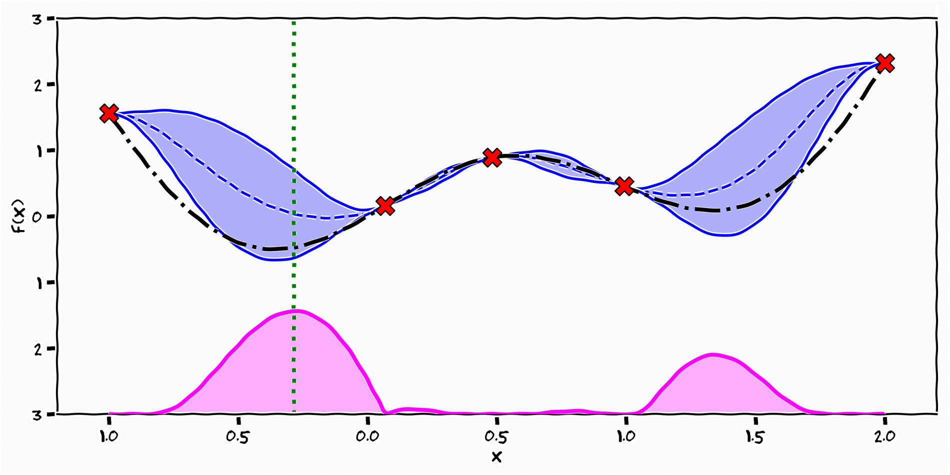
http://carlhenrik.com

$$\text{Data} + \text{Model} \overbrace{\rightarrow}^{Compute} \text{Prediction}$$

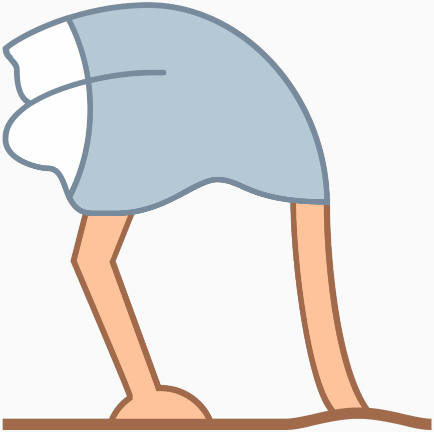
The role of Uncertainty/Ignorance

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

Bayesian Optimisation



What do we do with uncertainty?



Uncertainty Quantification

Aleatoric/Stochastic "Randomness" inherent in system, or noise in our measurement of system

Aleatoric/Stochastic "Randomness" inherent in system, or noise in our measurement of system

Epistemic Uncertainty related to our ignorance of a the underlying system

Uncertainty Quantification

Aleatoric/Stochastic "Randomness" inherent in system, or noise in our measurement of system

Epistemic Uncertainty related to our ignorance of a the underlying system

Computational Uncertainty related to finite computation, or intractable computations

"The need for probability only arises out of uncertainty: It has no place if we are certain that we know all aspects of a problem. But our lack of knowledge also must not be complete, otherwise we would have nothing to evaluate. There is thus a spectrum of degrees of uncertainty. While the probability for the sixth decimal digit of a number in a table of logarithms to equal 6 is 1/10 a priori, in reality, all aspects of the corresponding problem are well determined, and, if we wanted to make the effort, we could find out its exact value. The same holds for interpolation, for the integration methods of Cotes or Gauss, etc"

– Henri Poincare, 1896

Computational Uncertainty

- Computation is expensive, how much knowledge will I gain from computing more?

Computational Uncertainty

- Computation is expensive, how much knowledge will I gain from computing more?
- What should I compute in order to reduce my uncertainty as much as possible?

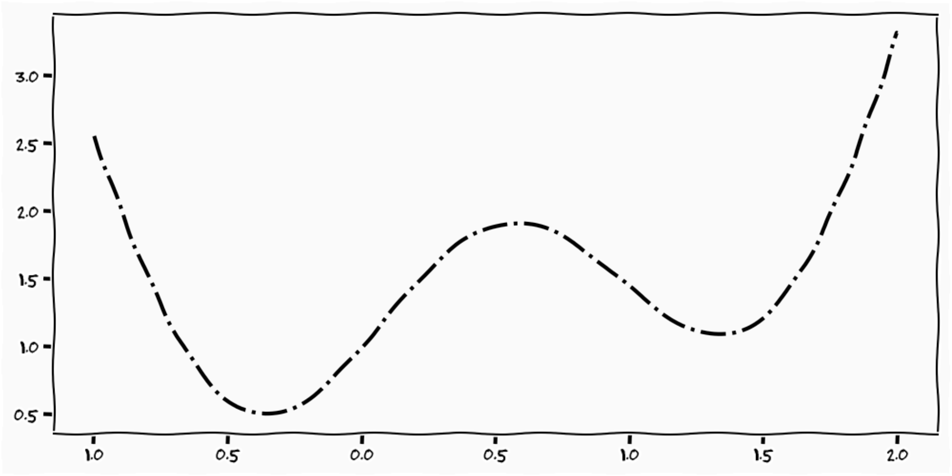
Computational Uncertainty

- Computation is expensive, how much knowledge will I gain from computing more?
- What should I compute in order to reduce my uncertainty as much as possible?
- How much should I trust the computation I have done?

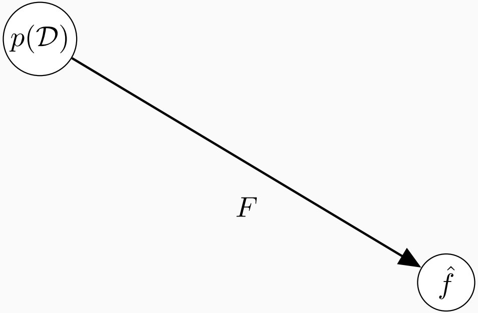
Computational Uncertainty

- Computation is expensive, how much knowledge will I gain from computing more?
- What should I compute in order to reduce my uncertainty as much as possible?
- How much should I trust the computation I have done?
- How precise should I do down-stream tasks based on the information in a computation?

I believe in . . .

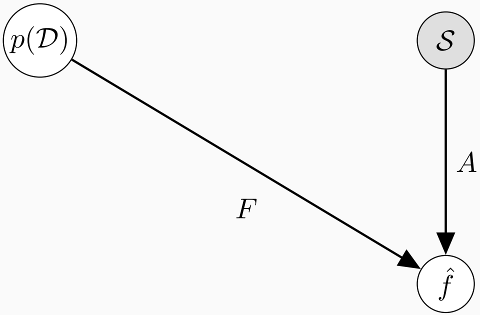

3.0
2.5
2.0
1.5
1.0
0.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0

Formalisation [Cockayne et al., 2017]



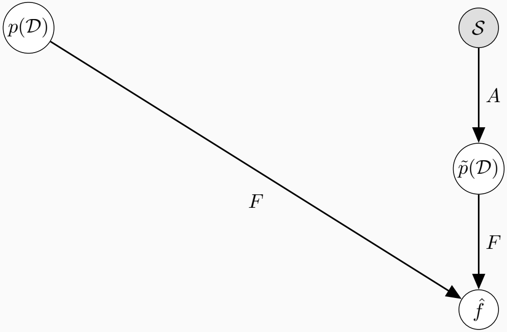
$$F : p(\mathcal{D}) \to p(\mathcal{Y}|\mathcal{X})$$

Formalisation [Cockayne et al., 2017]



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

Formalisation



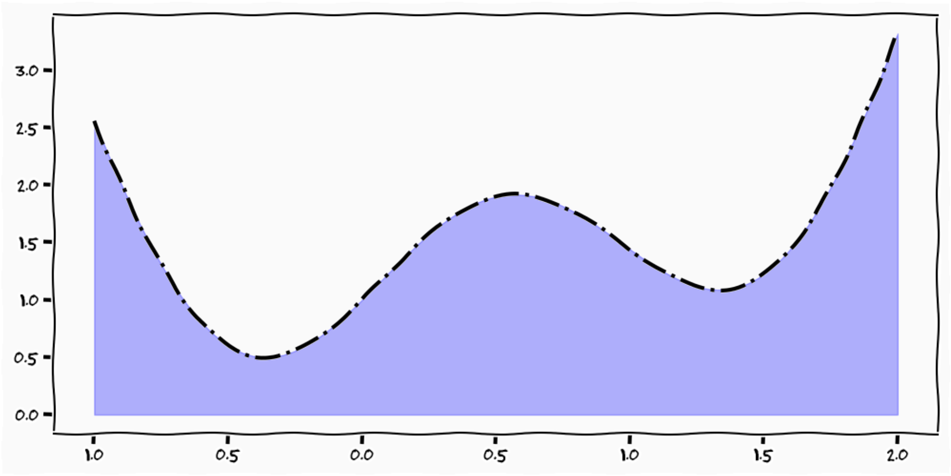
$$A \circ S \circ p(\mathcal{D}) \approx p(\mathcal{D})$$

Formalisation [Cockayne et al., 2017]

$$
\begin{tikzcd}
p(\mathcal{D}) \arrow[r, "S"] \arrow[dr, "F"'] & \mathcal{S} \arrow[d, "A"] \\
& \hat{f}
\end{tikzcd}
$$

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

Quantity of Interest



Integration is a significant numerical problem in many fields of science and engineering. It is a key step in inference, where it is encountered when averaging over the many states of the world consistent with observed data. Indeed, a provocative Bayesian view is that integration is the single challenge separating us from systems that fully automate statistics. More speculatively still, such systems may even exhibit artificial intelligence (ai).
– Hennig, Osbourne, Kersting

Integration

$$F := \int f(x) \mathrm{d}\nu(x)$$

- $\nu(x)$ is the measure that we are integrating over

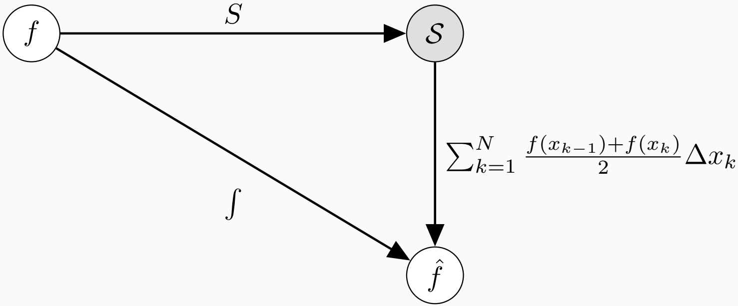
Integration

$$\underbrace{p(\mathcal{D})}_{F} = \int \underbrace{p(\mathcal{D} \mid \theta)}_{f(\theta)} \underbrace{p(\theta)\mathrm{d}\theta}_{\mathrm{d}\nu(\theta)}$$

- marginalisation[1] is integration over the prior probability measure on the parameter

[1] think of computing the evidence

Quadrature [Cockayne et al., 2017]



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

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.

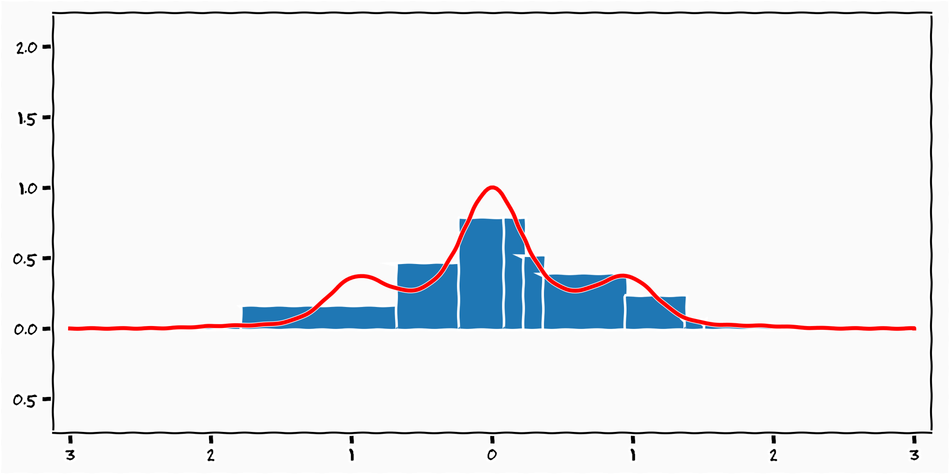
Numerical Computation

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

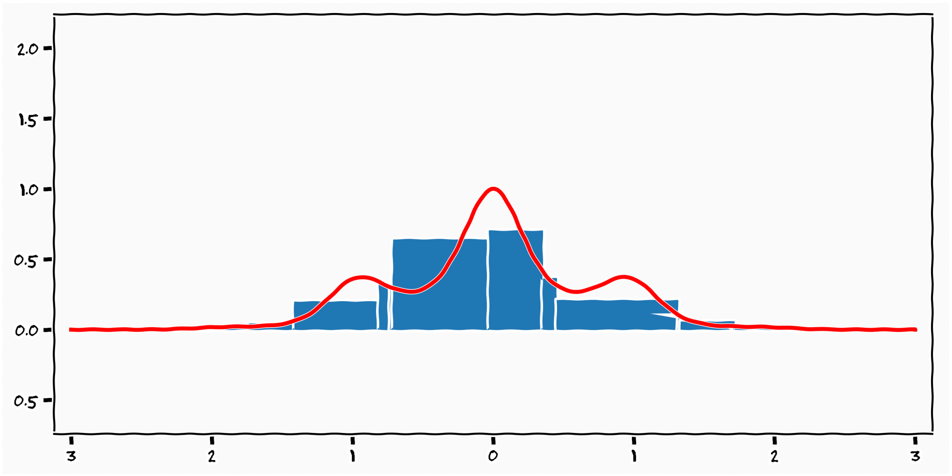
$\frac{\text{Numerical algorithms}}{\text{Statistical inference}}$ takes data in the form of $\frac{\text{evaluations of computations}}{\text{measurements of observed variables}}$ and aims to return predictions of the quantity of interest.

Should we think about computation as inference?

Quadrature



Quadrature



Decision which algorithm to use when

Use of Computational Uncertainty

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Use of Computational Uncertainty

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Decision when to stop computation

Use of Computational Uncertainty

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Decision when to stop computation

Decision effect on downstream tasks

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

When computation was expensive

Albert Valentionvic Suldin (1924-1996) worked on error minimising estimators for numerical algorithms, how to design algorithms from a statistical perspective

When computation was expensive

Albert Valentionvic Suldin (1924-1996) worked on error minimising estimators for numerical algorithms, how to design algorithms from a statistical perspective

Frederick Michael Larkin (1936-1982) incorporating the notion of prior knowledge into numerical algorithms to make robust calculations

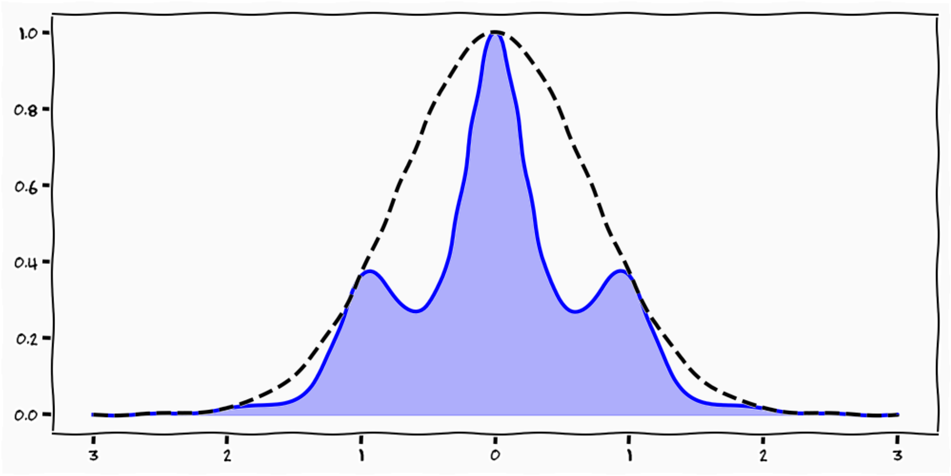
Bayesian Quadrature

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

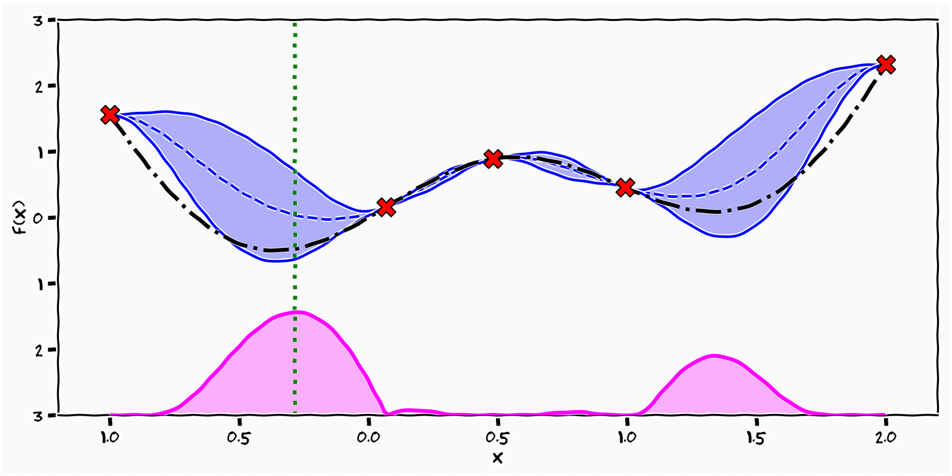


What we would like

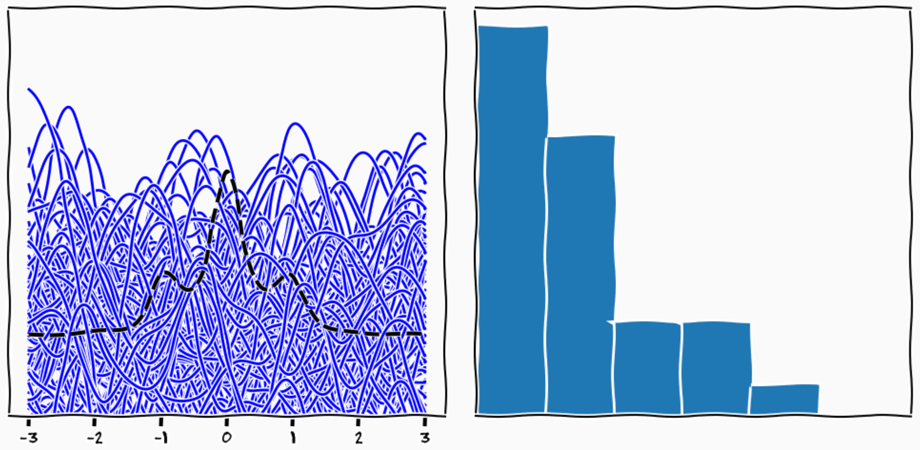
$$p(F \mid Y)$$

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

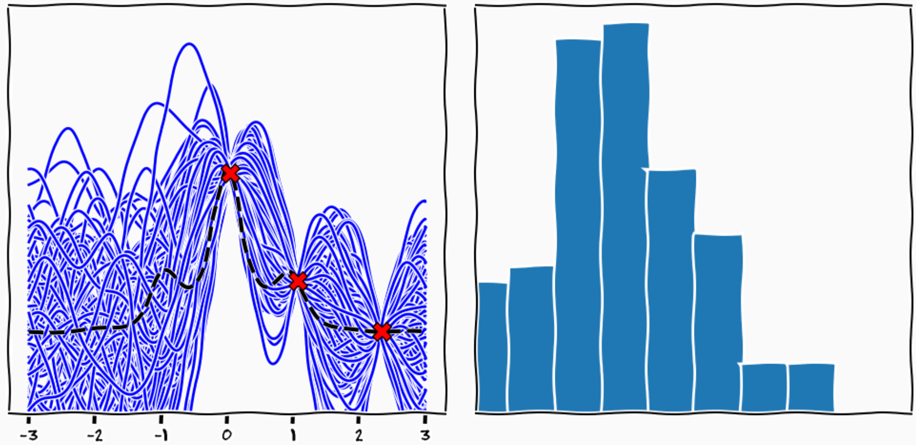
Emulation



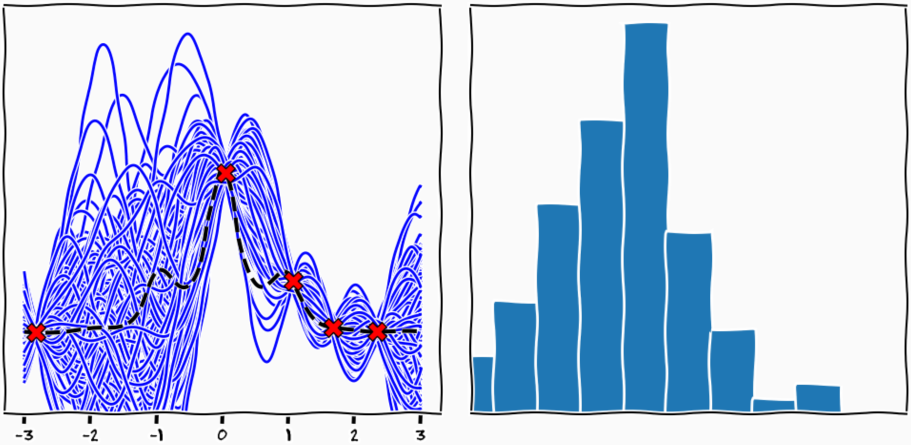
Quadrature



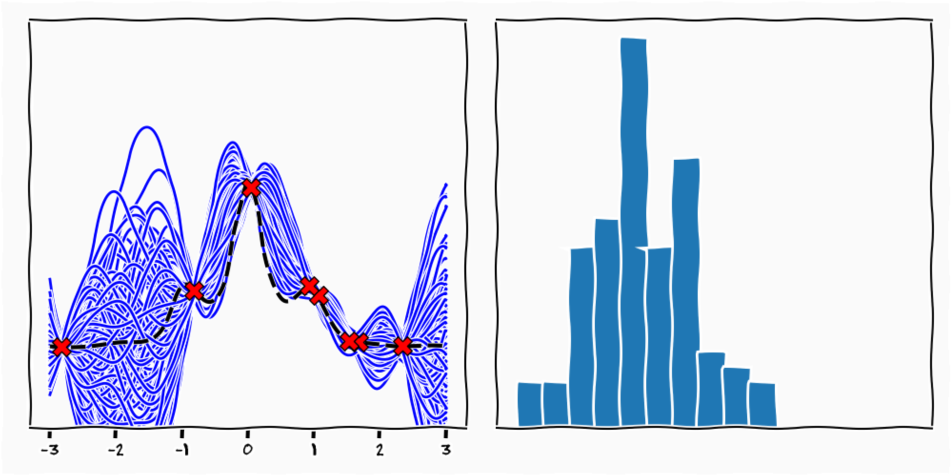
Quadrature



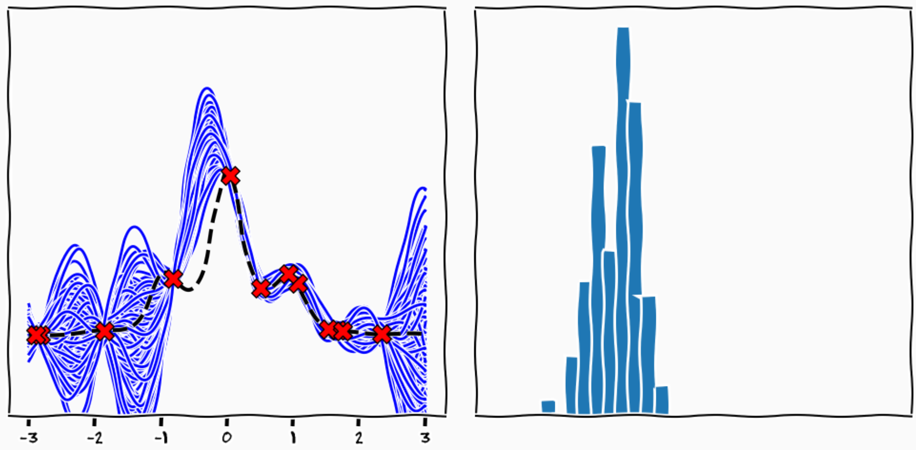
Quadrature



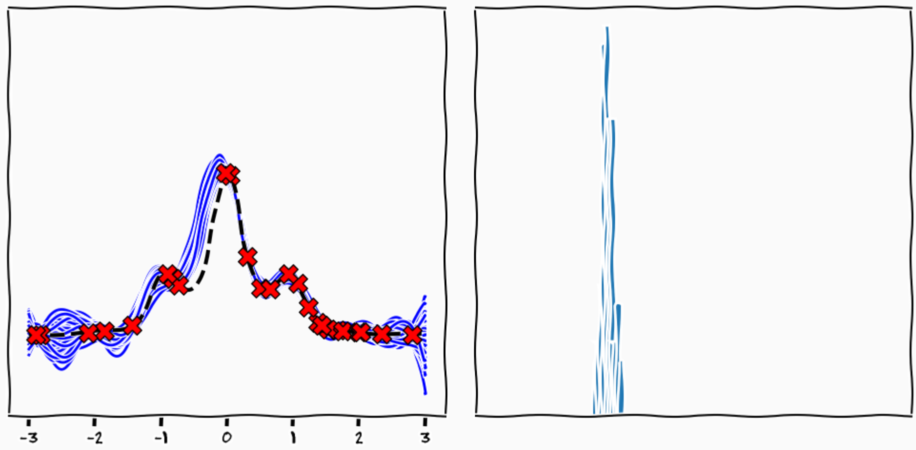
Quadrature



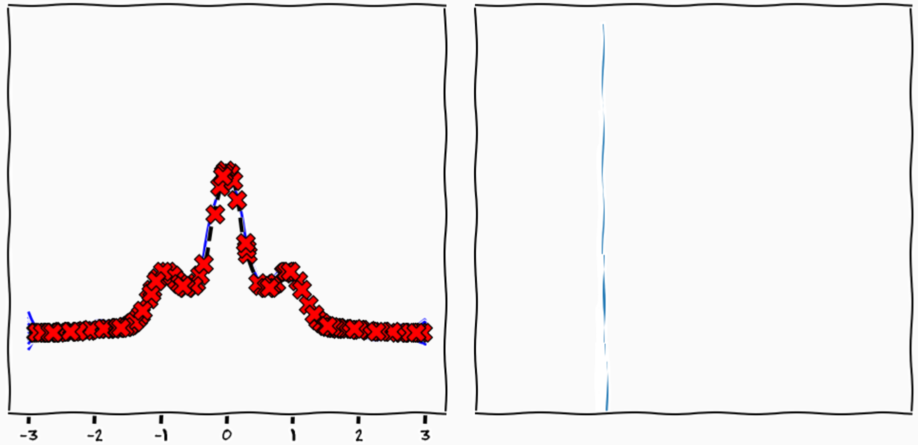
Quadrature



Quadrature



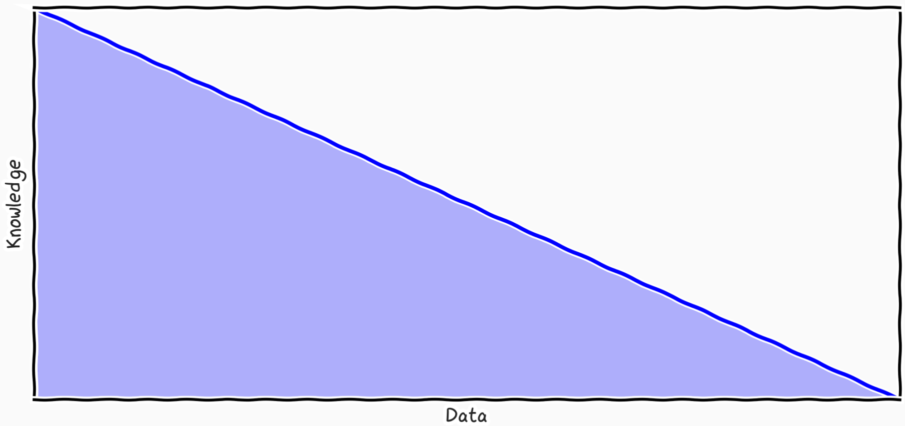
Quadrature



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

Knowledge

- $f(x)$ strictly positive $\Rightarrow F > 0$
- bounded above by,
$$f(x) \leq e^{-x^2}$$
- Therefore,
$$0 < F < \int_{-\infty}^{\infty} e^{-x^2} \mathrm{d}x = \sqrt{\pi}$$


Knowledge
Data

Integration

$$F :- = \int f(x) \mathrm{d}\nu(x)$$

- $\nu(x)$ is the measure that we are integrating over

Bayesian Quadrature [O'Hagan, 1991]

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

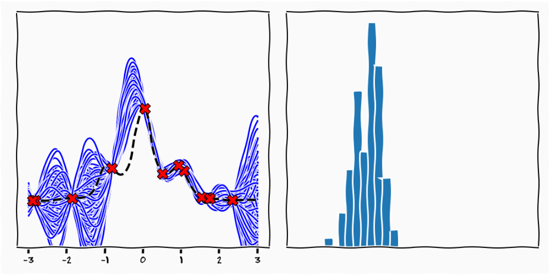
Bayesian Quadrature [O'Hagan, 1991]

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

Bayesian Quadrature [O'Hagan, 1991]

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

- 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

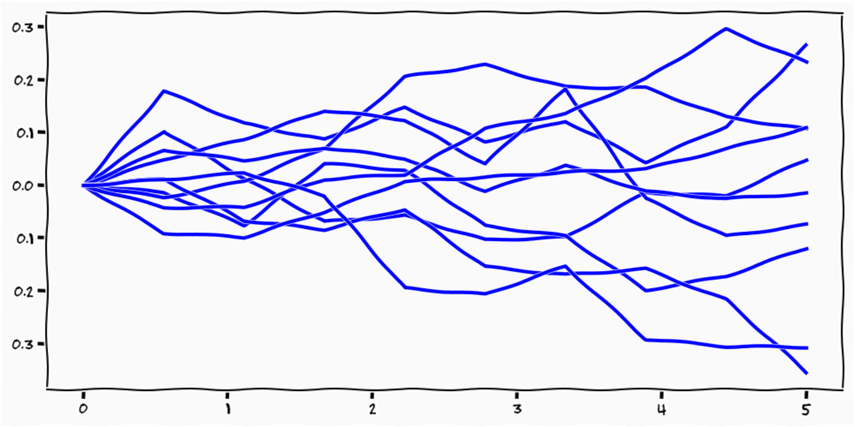


Integrand variance $\alpha(x) = k(x, x)$

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!!!!

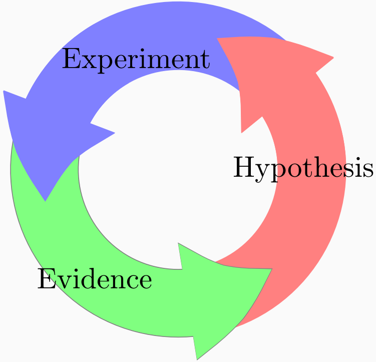
Quadrature Rule

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

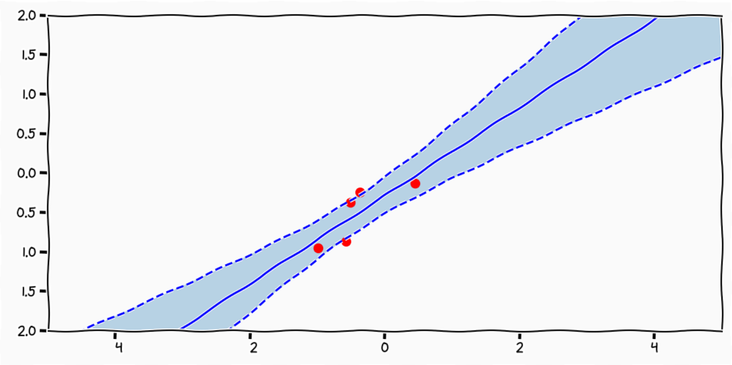
- This is the normal trapezoid rule!!!
- The algorithm is now tied to the function!!!!
- We can do inference over where to sample!!!!!!!!

Definition (Trapedzoid Rule)
The trapezoidal rule is the posterior mean estimate for the integral $F = \int_a^b f(x)\mathrm{d}x$ under any centred Wiener process prior $p(f) = \mathcal{GP}(0, k)$ with $k(x, x) = \theta^2(\min(x, x') - \kappa$

The Scientific Principle



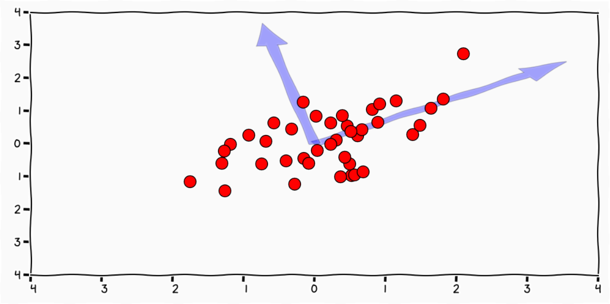
$$\text{Data} + \text{Model} \overbrace{\longrightarrow}^{Compute} \text{Prediction}$$



Legendre (1805) algorithm that reduces "error"

Gauss (1809) statistical model assuming i.i.d. Gaussian noise

Factor Analysis



Spearman (1904) proposed an algorithm to extract "factors" from data
Spearman, 1904

Hotelling (1936) concept of factor is clearly defined through a statistical model Hotelling, 1933

*

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

- 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?*

Summary

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]

[3] these thoughts have been around for a long time

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties

[3] these thoughts have been around for a long time

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]
- 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" therefore we should quantify our ignorance about its results

[3] these thoughts have been around for a long time

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]
- 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" therefore we should quantify our ignorance about its results
- Why?

[3]these thoughts have been around for a long time

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]
- 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" therefore we should quantify our ignorance about its results
- Why?
    - efficiency

[3] these thoughts have been around for a long time

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]
- 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" therefore we should quantify our ignorance about its results
- Why?
  - efficiency
  - down-stream tasks, uncertainty in computation should be part of decision

[3]these thoughts have been around for a long time

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[3]
- 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" therefore we should quantify our ignorance about its results
- Why?
  - efficiency
  - down-stream tasks, uncertainty in computation should be part of decision
  - learning/understanding algorithms in relation to problems/data

[3] these thoughts have been around for a long time

Numerical Computations[4]

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

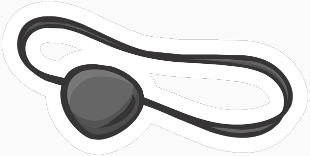
Linear Algebra given $As = y$ estimate x s.t. $Ax = b$

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

Analysis given $f(x, t)$ estimate $x(t)$ s.t. $\mathrm{d}x = f(x, t)$

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

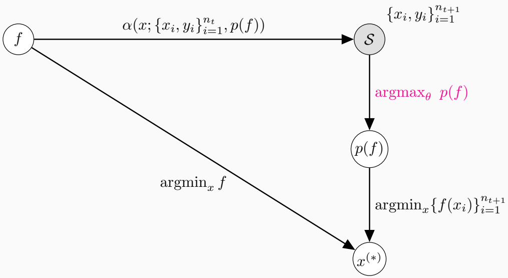
Assumptions: Algorithms



Statistical Learning

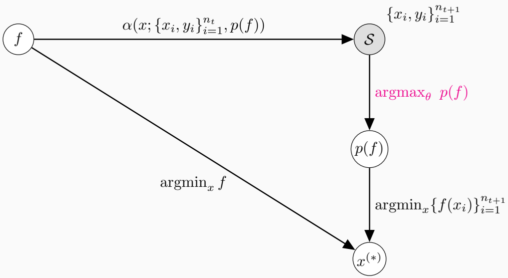
$\color{red}{\mathcal{A}}_{\mathcal{H}}(\mathcal{S})$

Is BO PN?



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

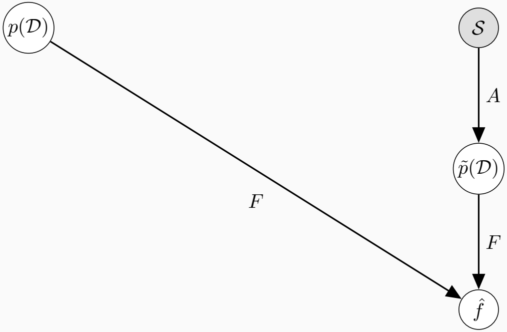
Is BO PN?



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.
- Hotelling, H (Sept. 1933). "Analysis of a complex of statistical variables into principal components.". In: *Journal of Educational Psychology* 24.6, pp. 417–441.

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.

Spearman, Charles (1904). “" General Intelligence," Objectively Determined and Measured”. In: *The American Journal of Psychology* 15.2, pp. 201–292.

Learning Hyper-parameters

$$\{\hat{\beta}, \hat{\ell}, \hat{\sigma}\} = \underset{\beta,\ell,\sigma}{\operatorname{argmax}} \ \log \int p(y \mid f)p(f)\mathrm{d}f$$

- Update the hyper-parameters of the GP
- We can do this by gradient based optimisation
  - *or with a Bayes Opt loop!*

[5] *. . . and here I was thinking that you guys had some principles in life*

Learning Hyper-parameters

$$\{\hat{\beta}, \hat{\ell}, \hat{\sigma}\} = \underset{\beta,\ell,\sigma}{\operatorname{argmax}} \; \log \int p(y \mid f)p(f)\mathrm{d}f$$

- Update the hyper-parameters of the GP
- We can do this by gradient based optimisation
  - *or with a Bayes Opt loop!*
- *Are we doing MLE*[5]*?*

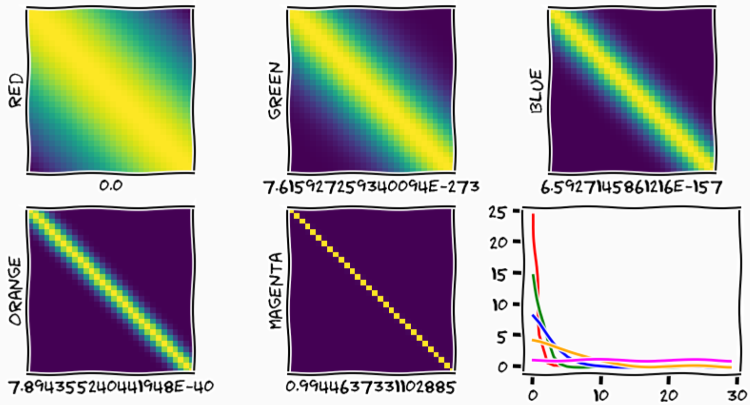
[5] *. . . .and here I was thinking that you guys had some principles in life*

Marginal Likelihood

$$
\begin{aligned}
\{\hat{\beta}, \hat{\ell}, \hat{\sigma}\} &= \underset{\beta,\ell,\sigma}{\operatorname{argmax}} \; \log \; \int p(y \mid f) p(f) \mathrm{d}f \\
&= \underset{\beta,\ell,\sigma}{\operatorname{argmin}} - \log \; p(y) \\
&= \underset{\beta,\ell,\sigma}{\operatorname{argmin}} \frac{1}{2} \mathrm{trace} \left(\mathbf{Y} \mathbf{K}^{-1} \mathbf{Y}^{\mathrm{T}} \right) + \frac{1}{2} \log |K| + \frac{N}{2} \log(2\pi)
\end{aligned}
$$

- $Data - fit$ *how well does the observations fit the model*
- $"Complexity"$ *how "smooth" is the functions*

Determinant



N

How to implement

jax.png

tensorflow.jpg

pyro_logo.png

Semantics

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

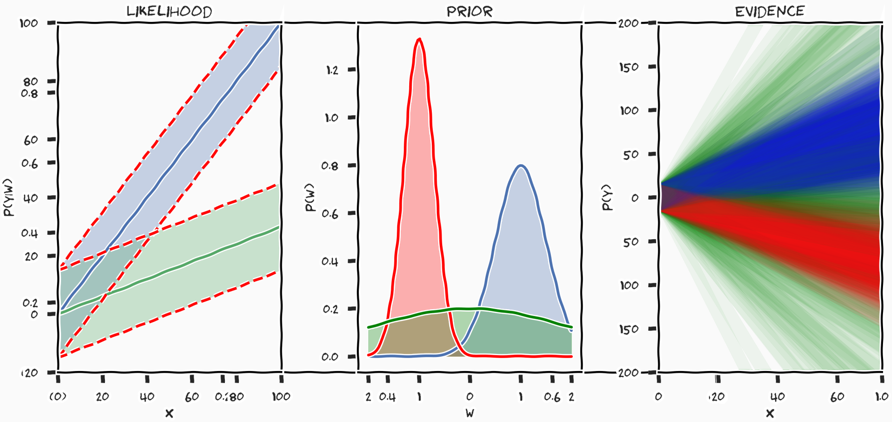
Likelihood How much evidence is there in the data for a specific hypothesis

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

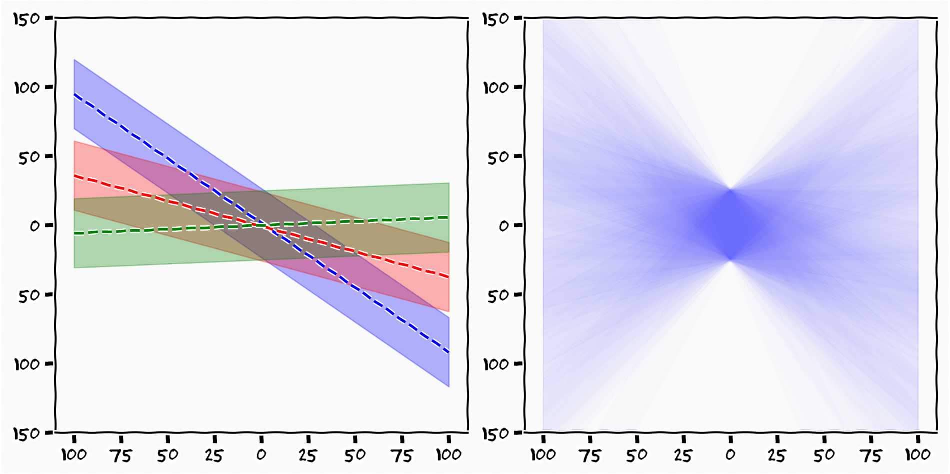
Linear Linear Model

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

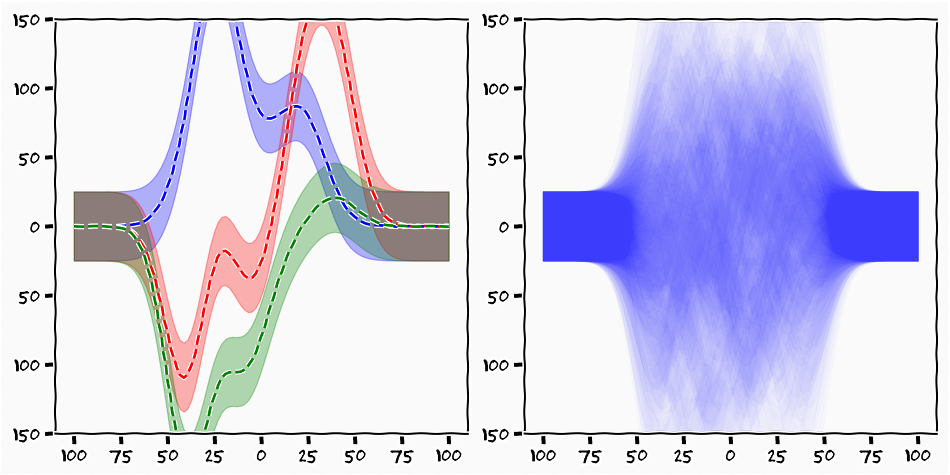
Basis function

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

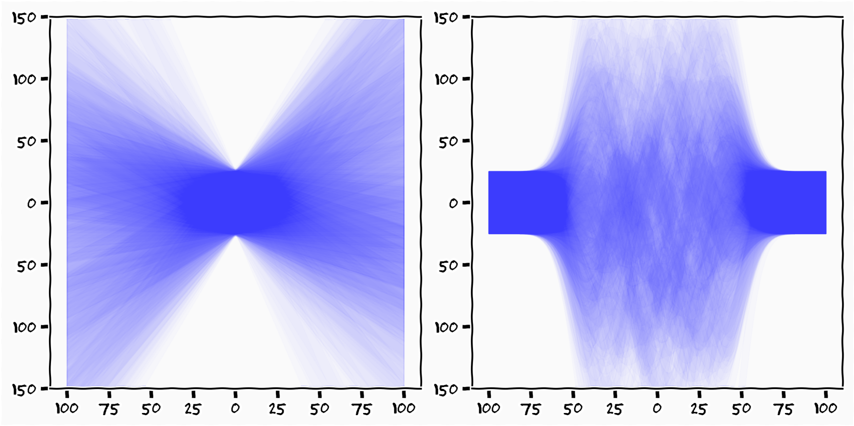
Linear Linear Regression



Linear Regression



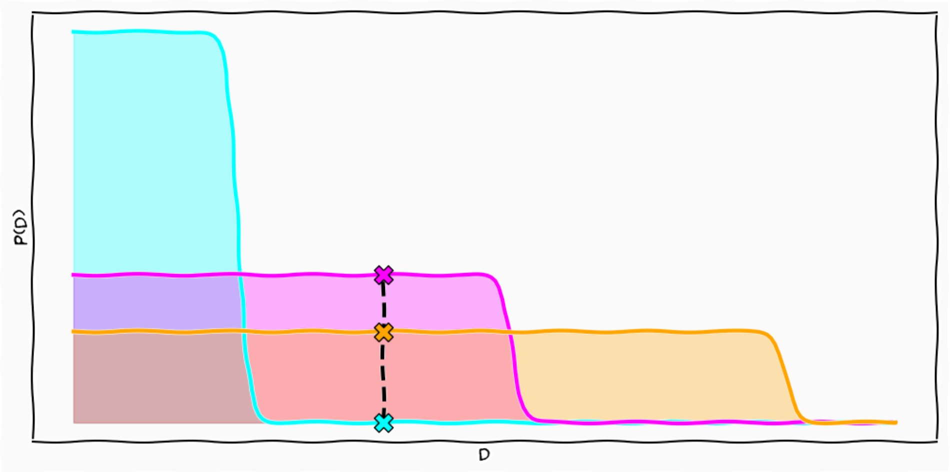
Evidence



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

Probabilities are a zero-sum game

The MacKay Plot Mackay, 1991



Occams Razor

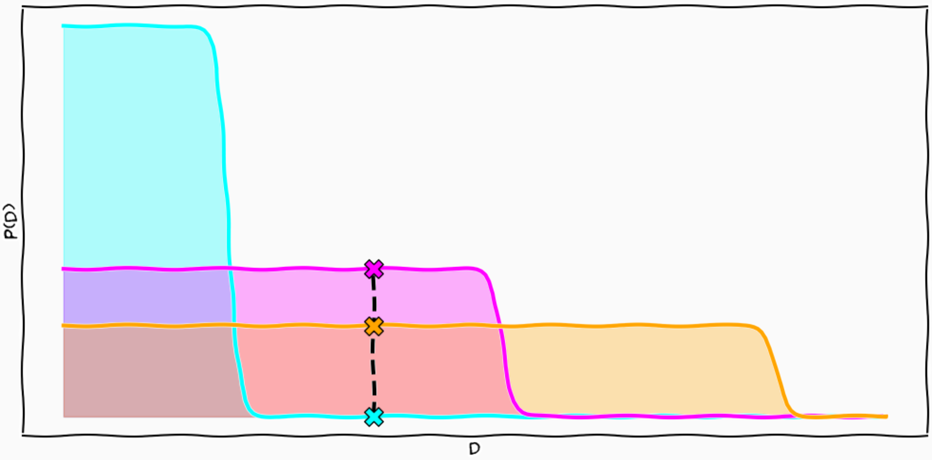
Definition (Occams Razor)
"All things being equal, the simplest solution tends to be the best one"

– William of Ockham

What is Simple?[6]

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

The MacKay Plot Mackay, 1991



Unsupervised Learning [Lawrence, 2005]

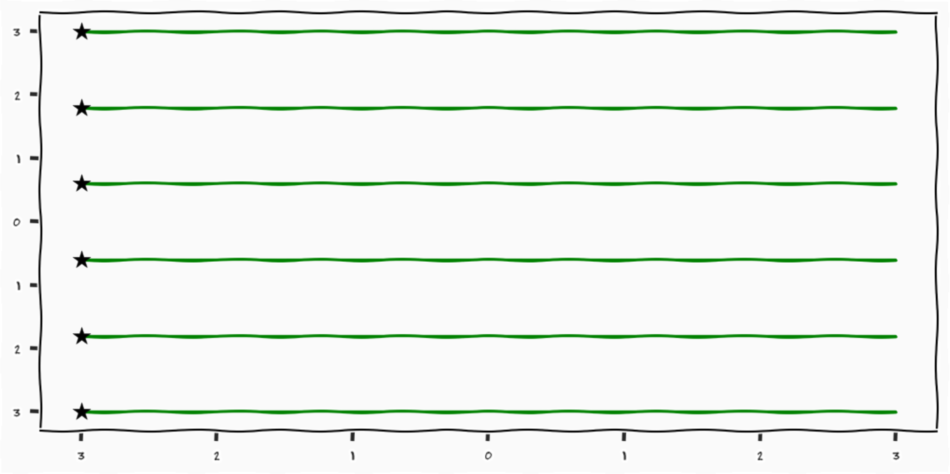
- Regression

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

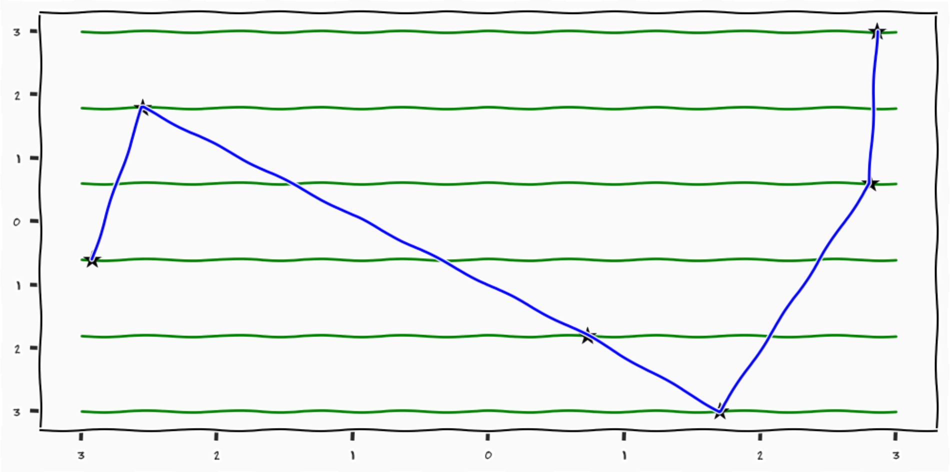
- "Unsupervised" Learning

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

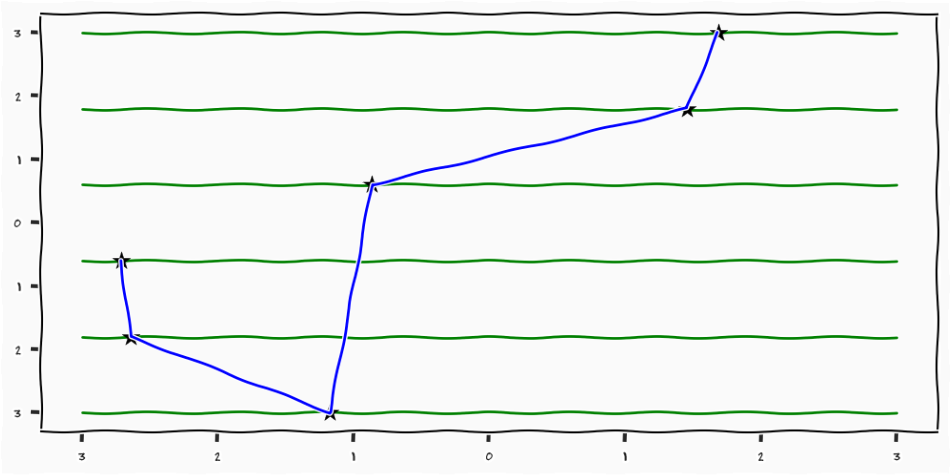
Unsupervised Learning



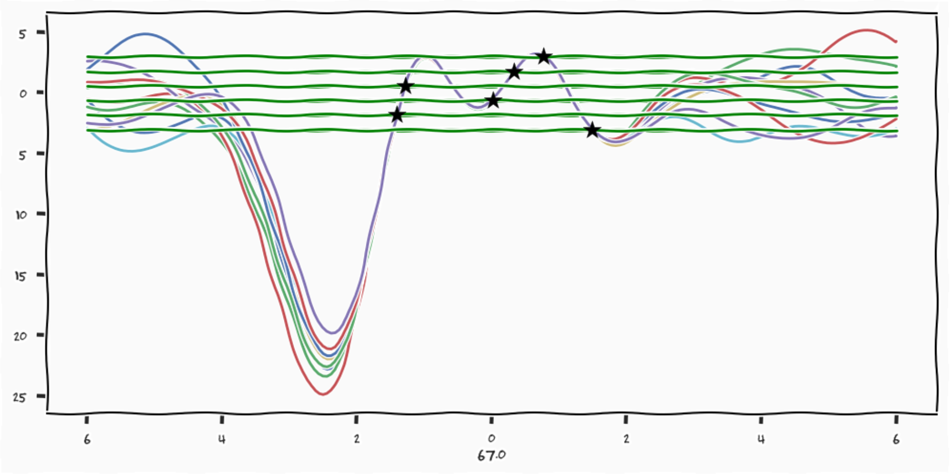
Unsupervised Learning



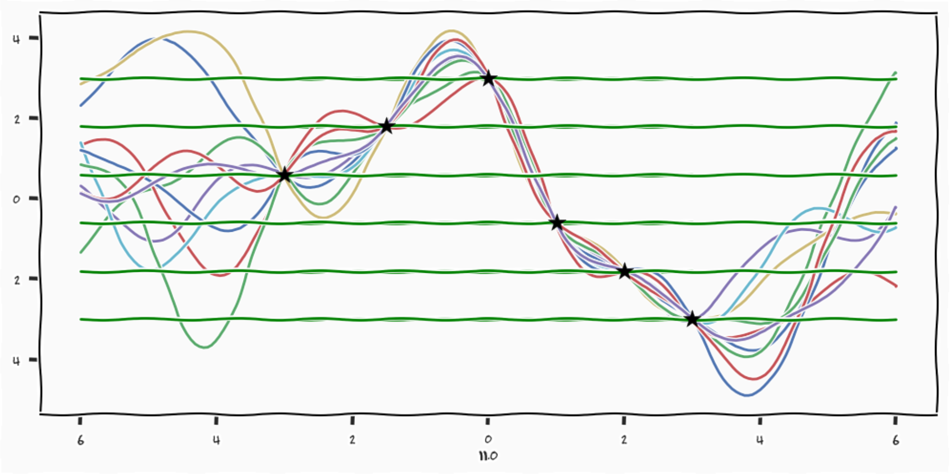
Unsupervised Learning



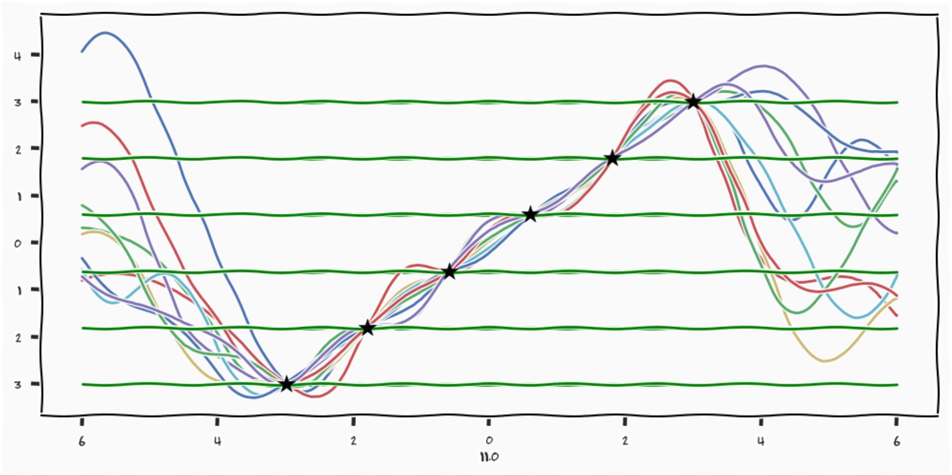
Gaussian Process Latent Variable Model [Lawrence, 2005]



Gaussian Process Latent Variable Model [Lawrence, 2005]



Gaussian Process Latent Variable Model [Lawrence, 2005]



Gaussian Process Latent Variable Model [Lawrence, 2005]