UNIVERSITY OF CAMBRIDGE

Machine Learning and the Physical World

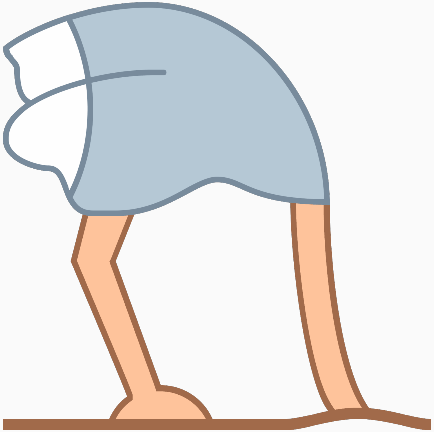
Lecture 9 : Probabilistic Numerics

Carl Henrik Ek - che29@cam.ac.uk

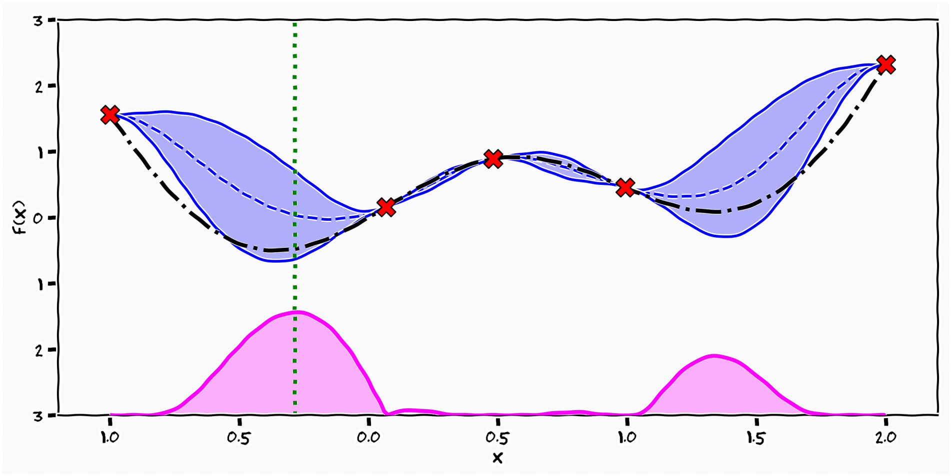
7th of November, 2024

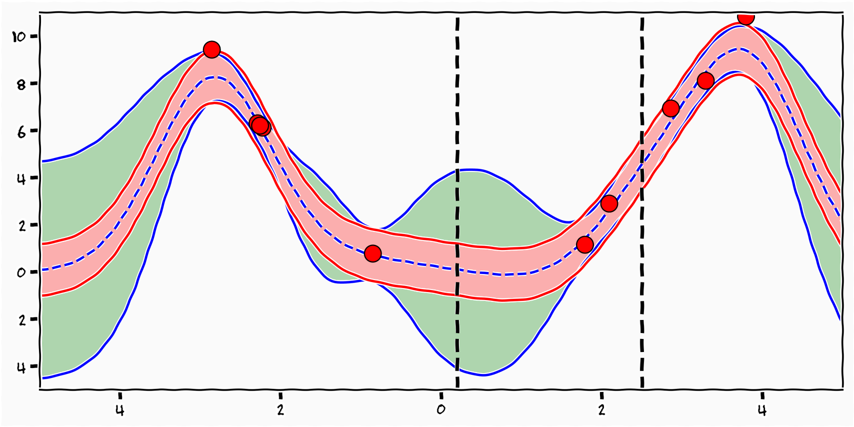
http://carlhenrik.com

What do we do with uncertainty?



Bayesian Optimisation





$$y_i = f_i + \epsilon$$

"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

Uncertainty Quantification

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

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*

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

Computational Decisions

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

Computational Decisions

- 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 Decisions

- 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?

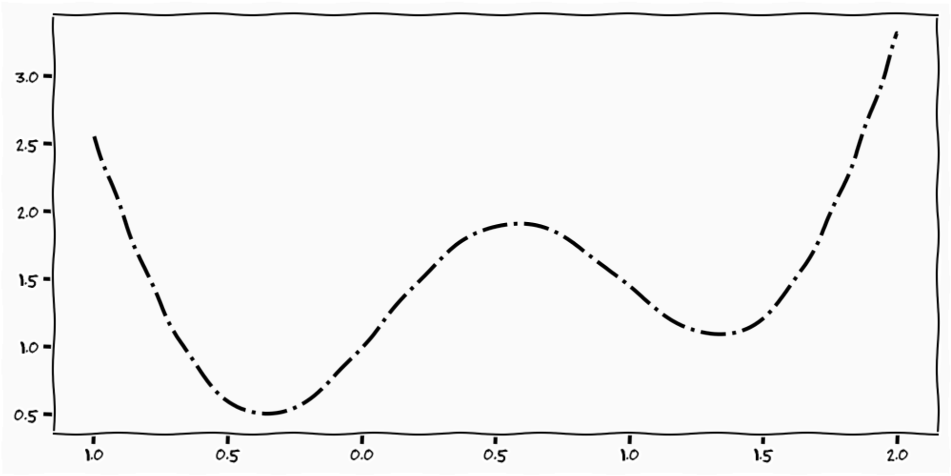
Computational Decisions

- 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 from a specific computation?

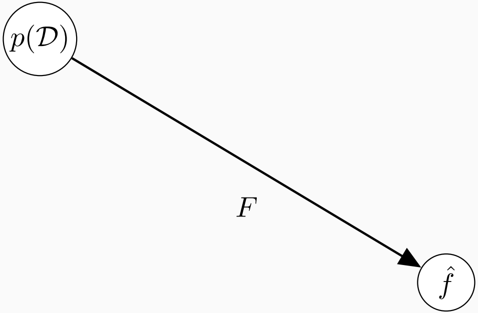
Why Probabilistic Numerics?

"[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

I believe in. . .

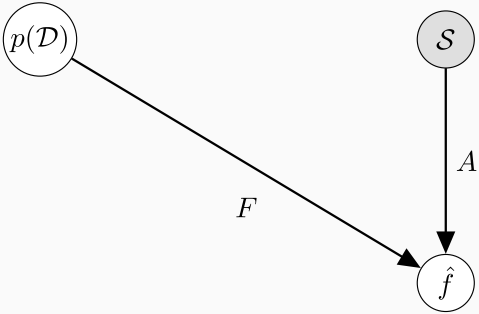

3.0
2.5
2.0
1.5
1.0
0.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0

Formalisation [Cockayne et al., 2017]



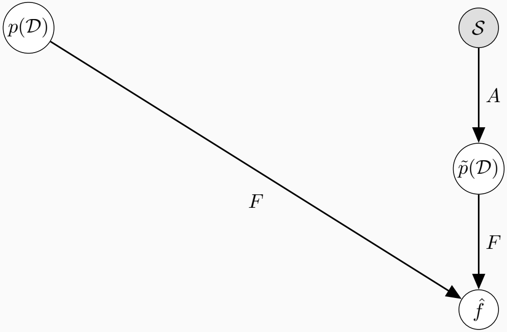
$$F : p(\mathcal{D}) \to p(\mathcal{Y}|\mathcal{X})$$

Formalisation [Cockayne et al., 2017]



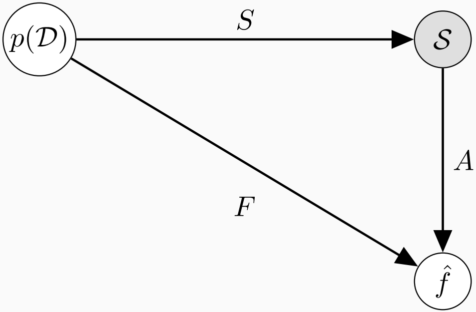
$$A \circ \mathcal{S} \approx F \circ p(\mathcal{D})$$

Formalisation

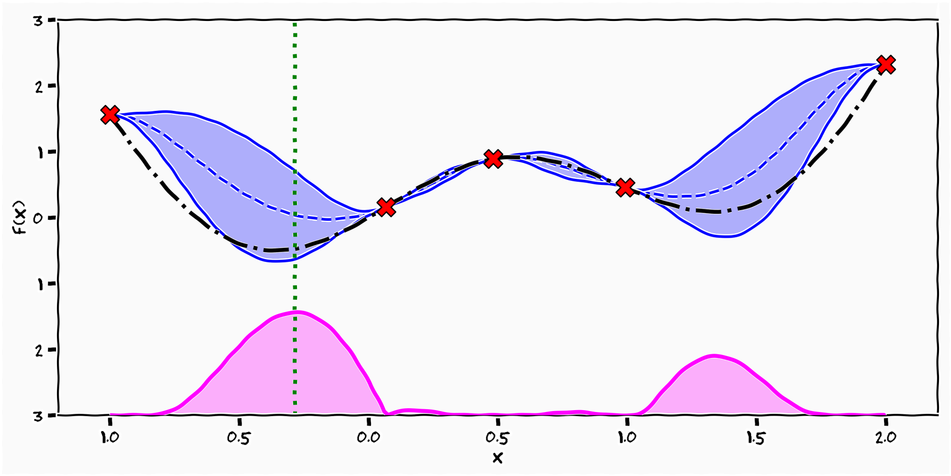


$$A \circ S \circ p(\mathcal{D}) \approx p(\mathcal{D})$$

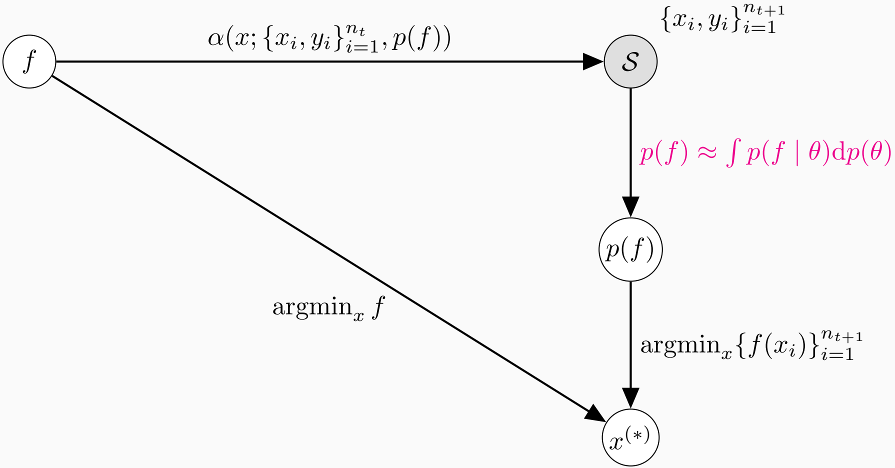
Formalisation [Cockayne et al., 2017]



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


3
2
1
0
1
2
3
f(x)
1.0
0.5
0.0
0.5
1.0
1.5
2.0
x

Bayesian Optimisation



Formalisation [Cockayne et al., 2017]

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

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

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

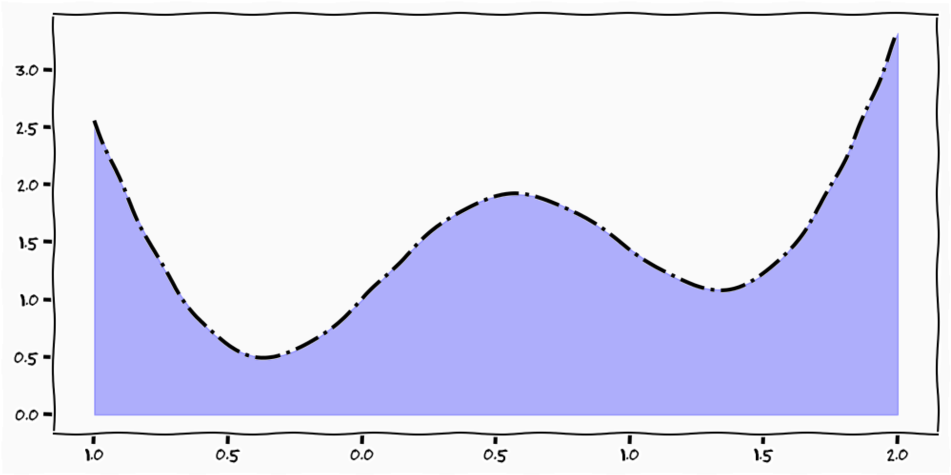
Optimisation given $\nabla f(x_i)$ estimate x s.t. $\nabla f(x) = 0$

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

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

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

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

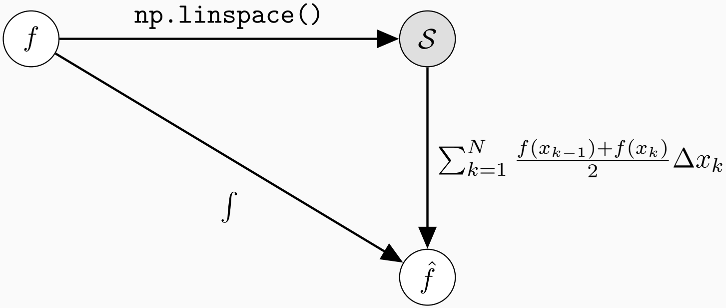
Integration

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

- marginalisation[2] is integration over the prior probability measure on the parameter

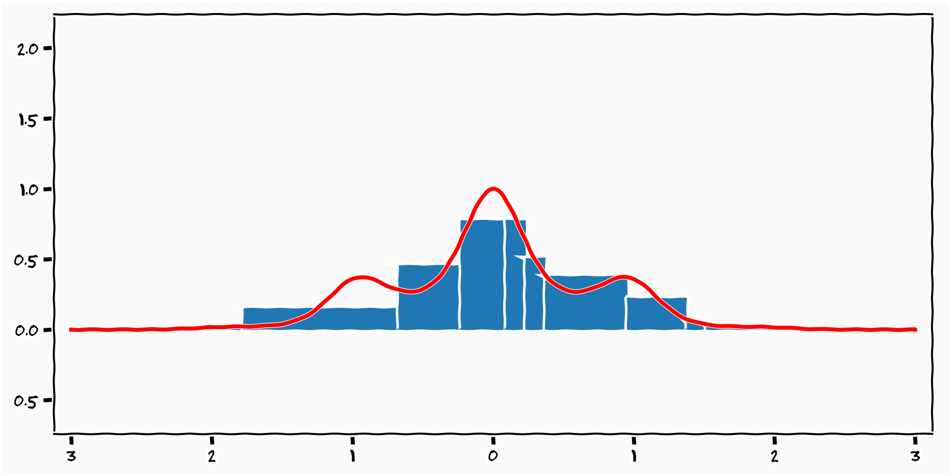
[2]think of computing the evidence

Quadrature [Cockayne et al., 2017]

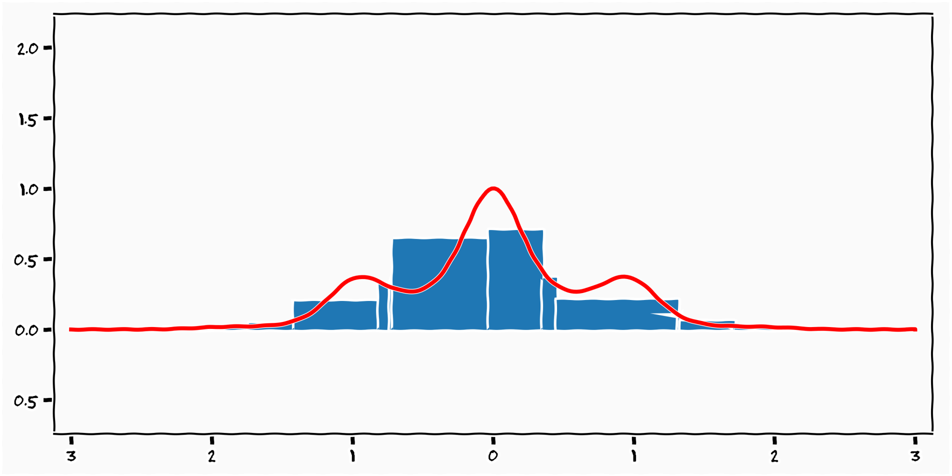


$$A \circ \mathcal{S} \approx \int f(x) \mathrm{d}x$$

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?

Use of Computational Uncertainty

$$p(\hat{f} \mid S, A)$$

Decision which algorithm to use when

Use of Computational Uncertainty

$$p(\hat{f} \mid S, A)$$

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Use of Computational Uncertainty

$$p(\hat{f} \mid S, A)$$

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Decision when to stop computation

Use of Computational Uncertainty

$$p(\hat{f} \mid S, A)$$

Decision which algorithm to use when

Decision efficient use of expensive algorithms

Decision when to stop computation

Decision effect on downstream tasks

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

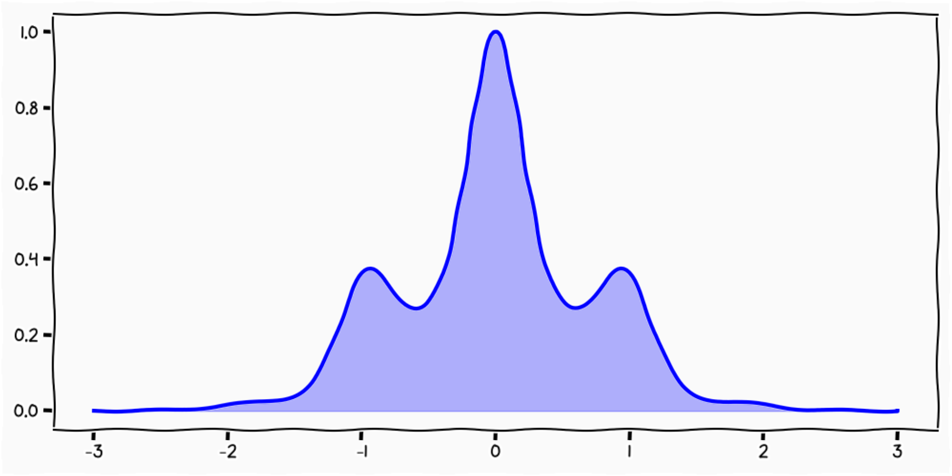
Bayesian Quadrature

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

What we would like

$$p(F \mid Y)$$

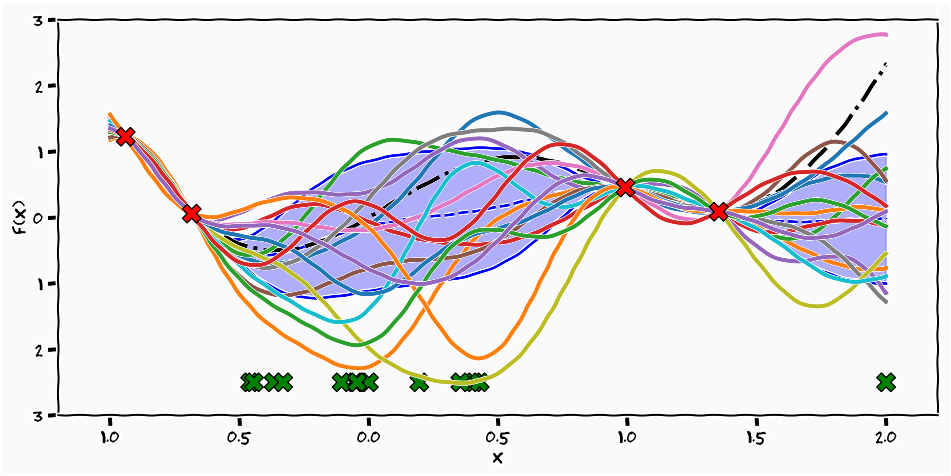
- given that I have seen data Y what is my belief about the integral
- allows for "active learning"

What we would like

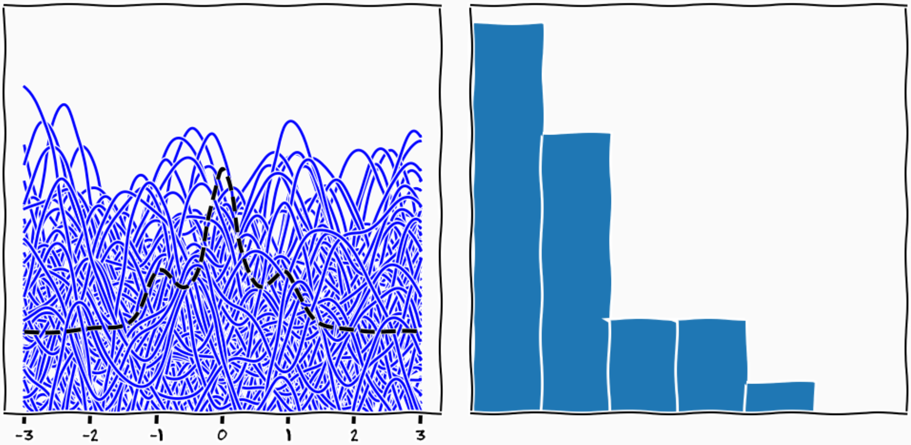
$$p(F \mid Y)$$

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

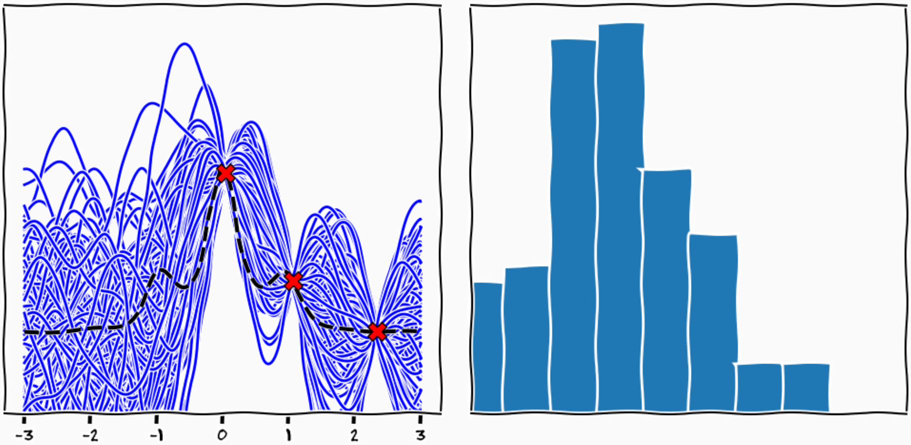
Emulation

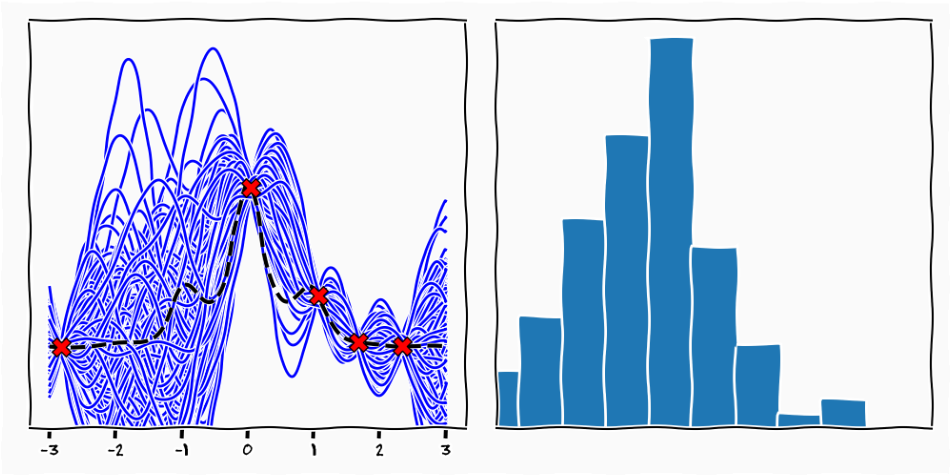


Quadrature

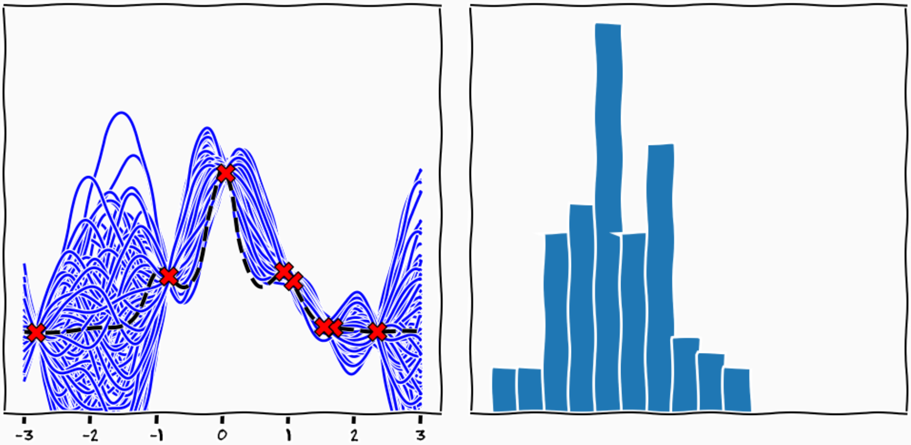


Quadrature

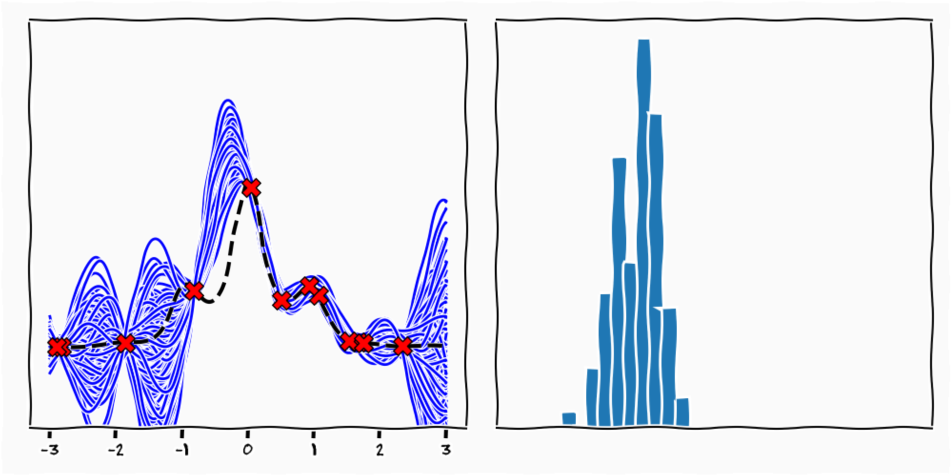



-3
-2
-1
0
1
2
3

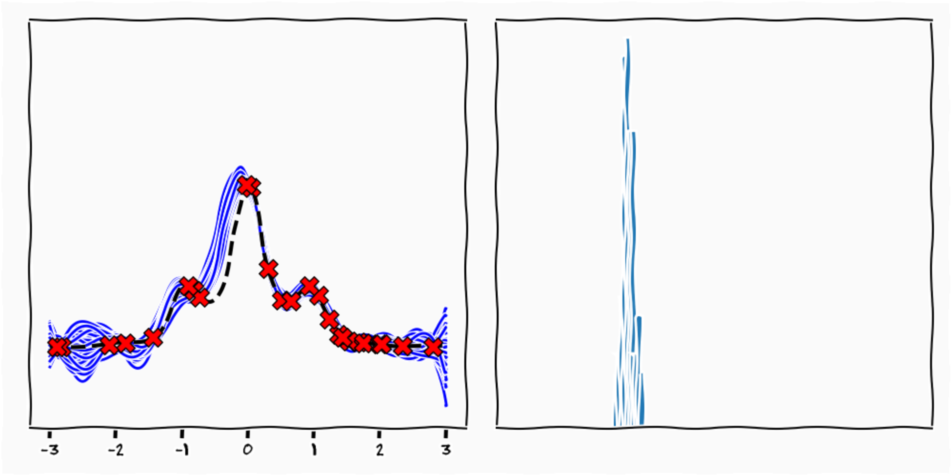
Quadrature



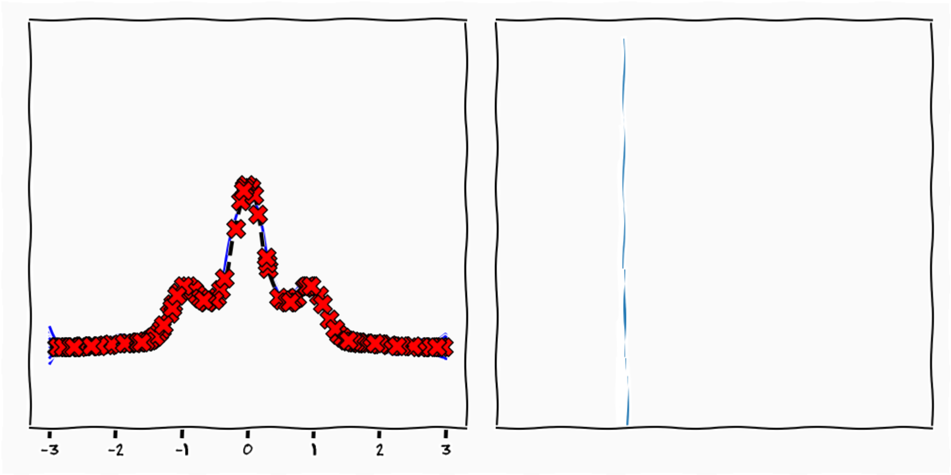
Quadrature



Quadrature



Quadrature

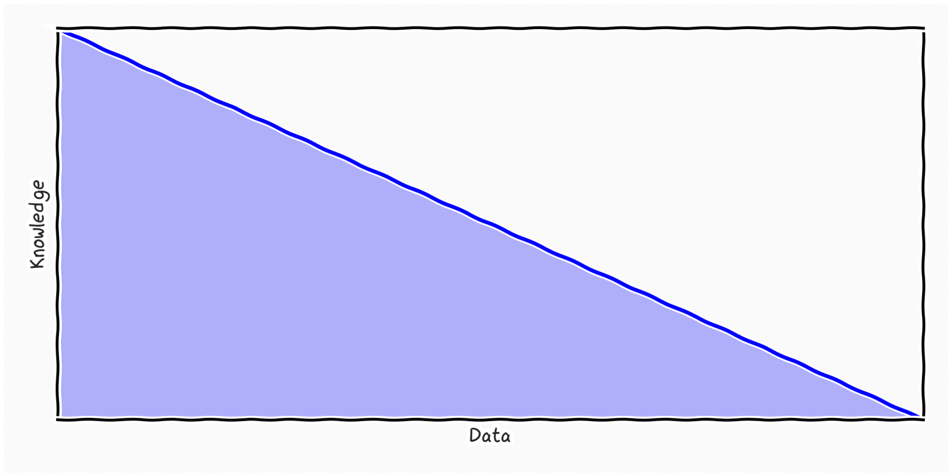


Model

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

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

Bayesian Quadrature [O'Hagan, 1991]

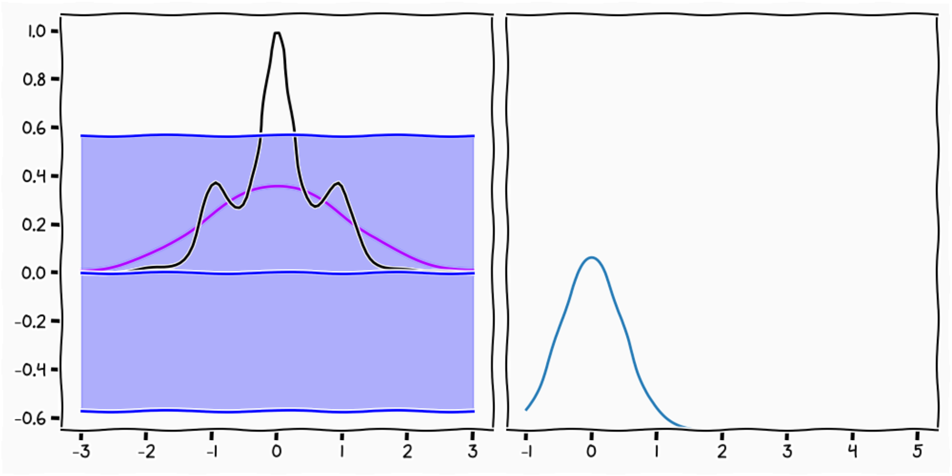
$$p\begin{pmatrix} Y \\ F \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_X \\ \int m_X(x)\mathrm{d}x \end{bmatrix}, \begin{bmatrix} k(X,X) & \int k(X,x)\mathrm{d}x \\ \int k(x,X)\mathrm{d}x & \int\int k(x,x')\mathrm{d}x\mathrm{d}x' \end{bmatrix}\right)$$

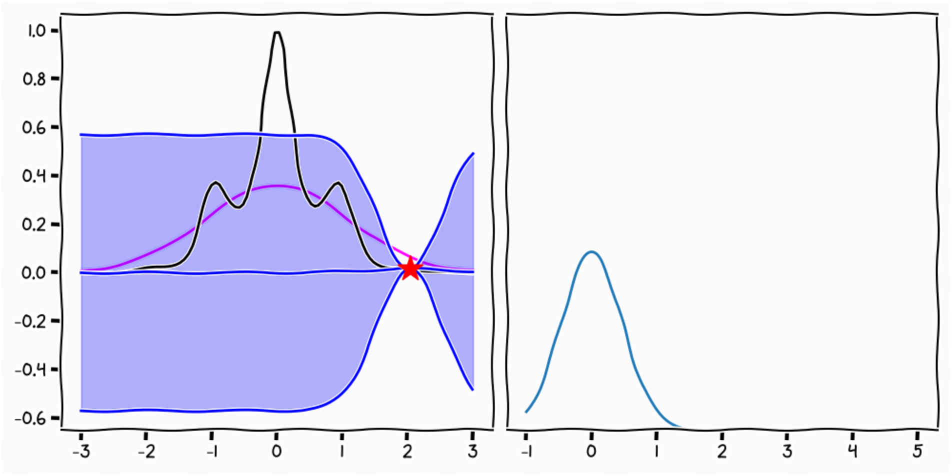
- We can derive $p(F \mid Y)$ through our normal conditioning procedure
- $p(F \mid Y) = \mathcal{N}(\mu_F, k_F)$ is a uni-variate Gaussian

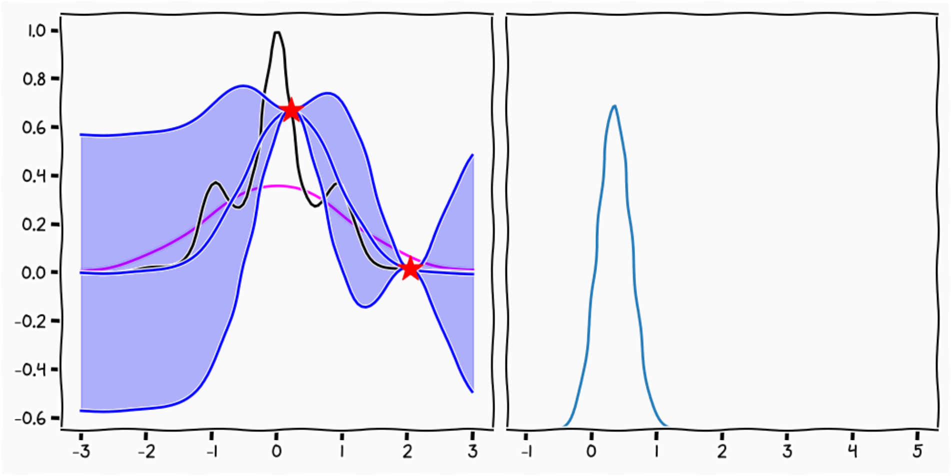
Bayesian Quadrature

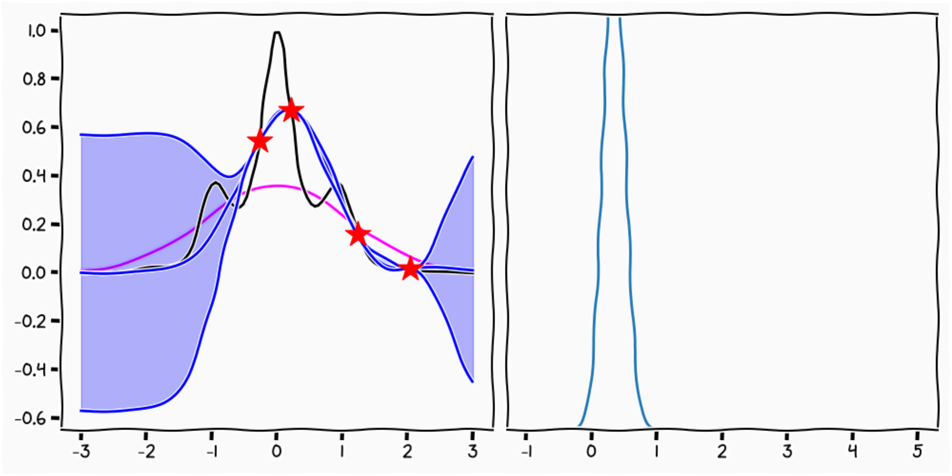
$$p\begin{pmatrix} Y \\ F \end{pmatrix} = \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_X \\ \int m_X(x)\mathrm{d}x \end{bmatrix}, \begin{bmatrix} k(X,X) & \int k(X,x)\mathrm{d}x \\ \int k(x,X)\mathrm{d}x & \int\int k(x,x')\mathrm{d}x\mathrm{d}x' \end{bmatrix}\right)$$

- 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


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-3
-2
-1
0
1
2
3
-1
0
1
2
3
4
5


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-3
-2
-1
0
1
2
3
-1
0
1
2
3
4
5


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-3
-2
-1
0
1
2
3
-1
0
1
2
3
4
5

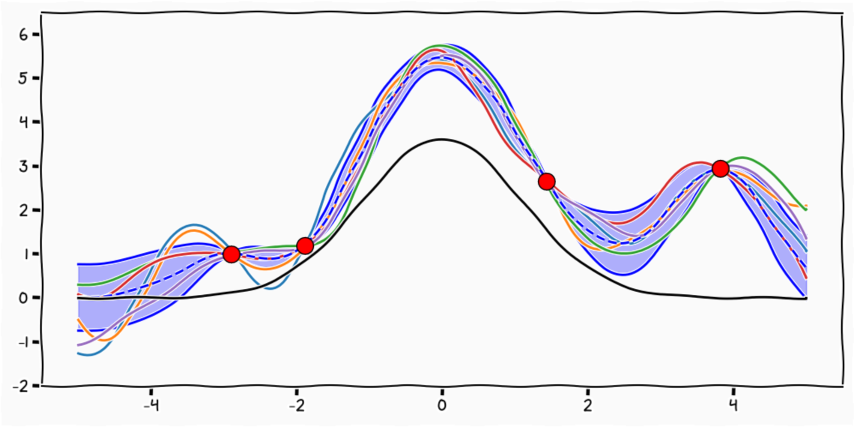

1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-3
-2
-1
0
1
2
3
-1
0
1
2
3
4
5

Bayesian Quadrature

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

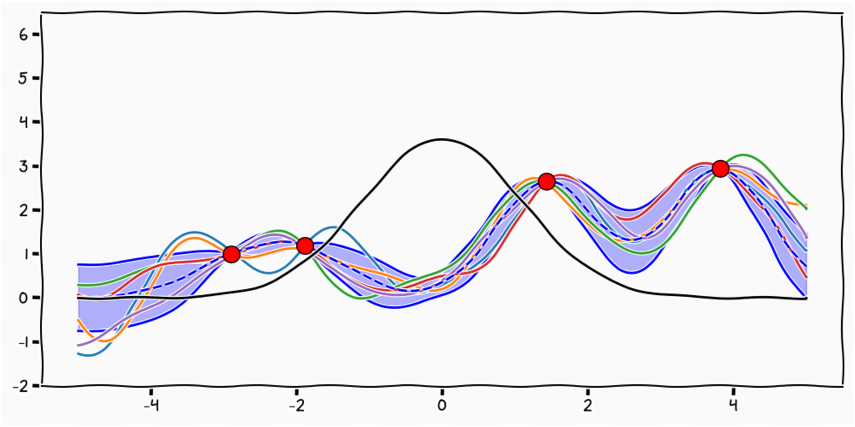
- We can derive $p(F \mid Y)$ through our normal conditioning procedure
- $p(F \mid Y) = \mathcal{N}(\mu_F, k_F)$ is a uni-variate Gaussian
- $p(Y \mid F) = \mathcal{N}(\mu_Y, k_Y)$ is a Gaussian process

Integral Constrained Samples



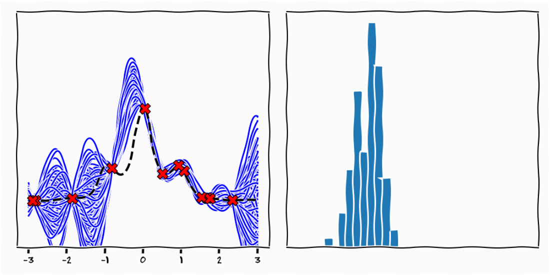
$$F = 4.0$$

Integral Constrained Samples



$$F = 1.0$$

Information Operator[3]

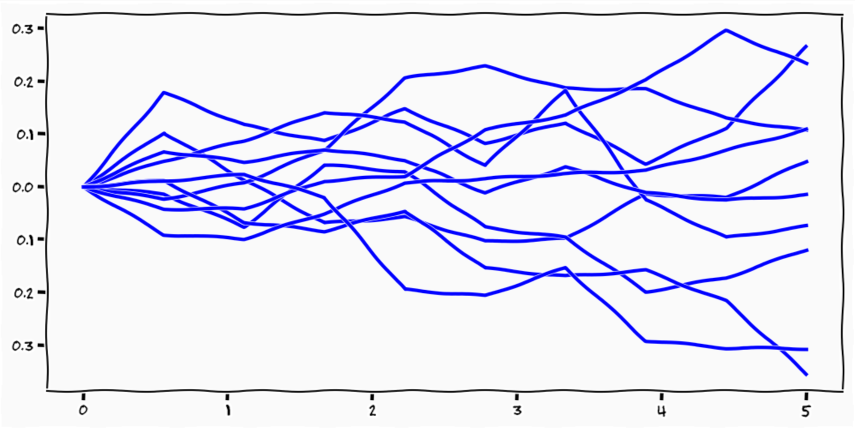


Integrand variance $\alpha(x) = k(x, x)$

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

[3] sometimes called a "Design Rule"

Choice of Covariance



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

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 our belief in the function!!!!

Quadrature Rule

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

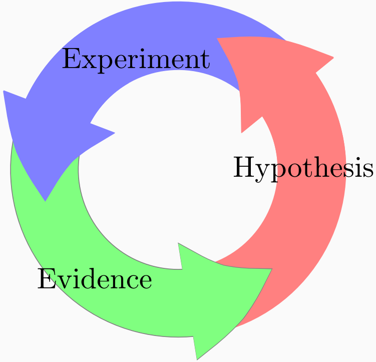
- This is the normal trapezoid rule!!!
- The algorithm is now tied to our belief in the function!!!!
- We can do inference over where to sample!!!!!!!!

Trapedzoid Rule

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 $k(x, x) = \theta^2(\min(x, x') - \kappa)$

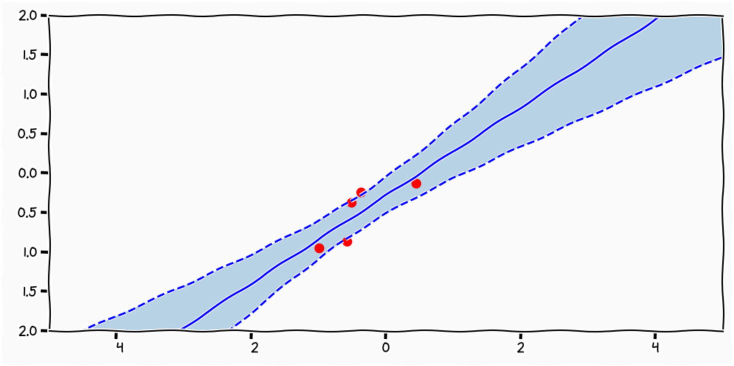

I'M NOT IMPRESSED
quickmeme.com

The Scientific Principle



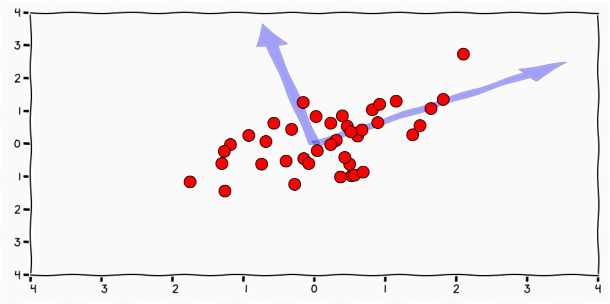
$$\text{Data} + \text{Model} \overbrace{\longrightarrow}^{Compute} \text{Prediction}$$

Least Squares Regression



Legendre (1805) algorithm that reduces "error"

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



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

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?*

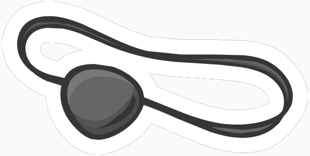
Neural Networks (Maybe) Evolved to Make Adam The Best Optimizer

In a talk, Olivier Bousquet has described the deep learning community as a giant genetic algorithm: Researchers in this community are exploring the space of all variants of algorithms and architectures in a semi-random way. Things that consistently work in large experiments are kept, the ones not working are discarded. .... the community is evolving only one set of parameters (architectures, initialization strategies, hyperparameters search algorithms, etc.) keeping most of the time the optimizer fixed to Adam.

"The Machine Learning Principle"[4]

"There is a notion of success . . . which I think is novel in the history of science. It interprets success as approximating unanalyzed data."
– Prof. Noam Chomsky

[4] Chomsky et al., 1980



Statistical Learning

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

Summary

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[5]

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

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[5]
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties

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

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[5]
- 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

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

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[5]
- 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?

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

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[5]
- 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

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

Summary

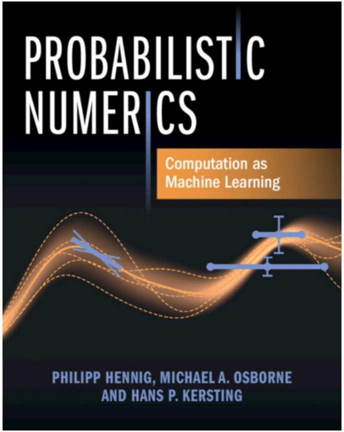
- Probabilistic Numerics extends the notion of statistical inference to computation[5]
- Computation is the process of extracting a latent property, machine learning is the statistical process of updating beliefs about latent properties
- Computation is often not "truth" therefore we should quantify our ignorance about its results
- Why?
  - efficiency
  - down-stream tasks, uncertainty in computation should be part of decision

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

Summary

- Probabilistic Numerics extends the notion of statistical inference to computation[5]
- 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

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



http://probnumschool.org

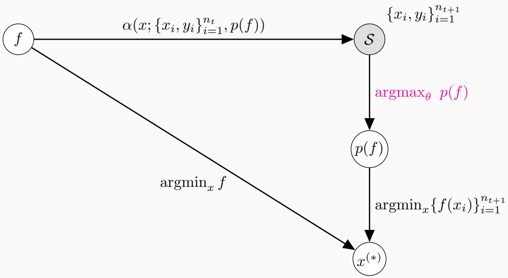
Adaptive probabilistic ODE solvers without adaptive memory requirements

Adaptive probabilistic solvers for ordinary differential equations (ODEs) have made substantial progress in recent years but can still not solve memory-demanding differential equations. In this talk, I review recent developments in numerically robust fixed-point smoothers and how to use them for constructing adaptive probabilistic ODE solvers. These new algorithms use drastically less memory than their predecessors and are the first adaptive probabilistic numerical methods compatible with scientific computing in JAX .

`SS03, 16-17`

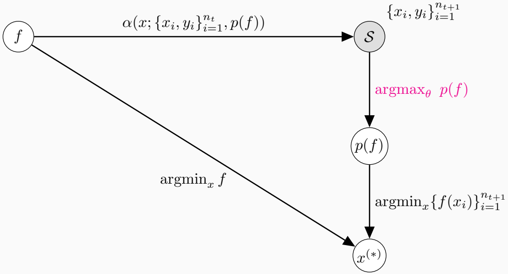
eof

Is BO PN?



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

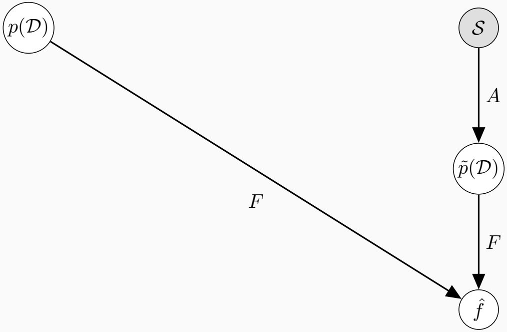
Is BO PN?



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

No the probabilistic model is not over the quantity of interest

Formalisation



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

References

Chomsky, Noam A and Jerry A Fodor (1980). **"The inductivist fallacy."** In: *Language and Learning: The Debate between Jean Piaget and Noam Chomsky*.

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.

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.