The computations of acting agents and the agents acting in computations

Philipp Hennig

ICERM

5 June 2017

Research Group for Probabilistic Numerics
Max Planck Institute for Intelligent Systems
Tübingen, Germany

MAX-PLANCK-GESELLSCHAFT

Emmy Noether-Programm
DFG Deutsche Forschungsgemeinschaft

Some of the presented work was supported by the Emmy Noether Programme of the DFG

Part I: The computations of acting agents — 09:00–09:45

- a minimal introduction to machine learning
- the computational tasks of learning agents
- some special challenges, some house numbers

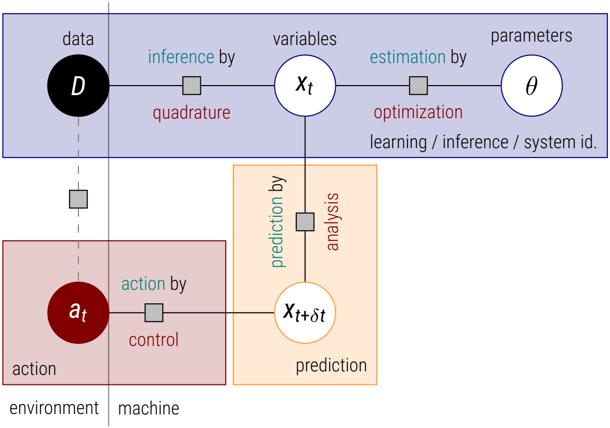
Part II: The agents acting in computations — 10:30–11:15

- computation is inference
- new challenges require new answers
- a computer science view on numerical computations

An Acting Agent

autonomous interaction with a data-source

from Hennig, Osborne, Girolami, Proc. Roy. Soc. A, 2015



The Very Foundation

probabilistic inference

$$p(x \mid D) = \frac{p(x)p(D \mid x)}{\int p(x)p(D \mid x)\, dx}$$

prior	explicit representation of assumptions about latent variables
likelihood	explicit representation of assumptions about generation of data
posterior	structured uncertainty over prediction
evidence	marginal likelihood of model

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\intercal \Sigma^{-1}(x-\mu)\right)$$

Gaussian Inference

the link between probabilistic inference and linear algebra

- products of Gaussians are Gaussians $\quad C := (A^{-1} + B^{-1})^{-1} \quad c := C(A^{-1}a + B^{-1}b)$

$$\mathcal{N}(x; a, A)\mathcal{N}(x; b, B) = \mathcal{N}(x; c, C)\mathcal{N}(a; b, A + B)$$

- marginals of Gaussians are Gaussians

$$\int \mathcal{N}\left[\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}\right] dy = \mathcal{N}(x; \mu_x, \Sigma_{xx})$$

- (linear) conditionals of Gaussians are Gaussians

$$p(x \mid y) = \frac{p(x,y)}{p(y)} = \mathcal{N}\left(x; \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\right)$$

- linear projections of Gaussians are Gaussians

$$p(z) = \mathcal{N}(z; \mu, \Sigma) \quad \Rightarrow \quad p(Az) = \mathcal{N}(Az, A\mu, A\Sigma A^\intercal)$$

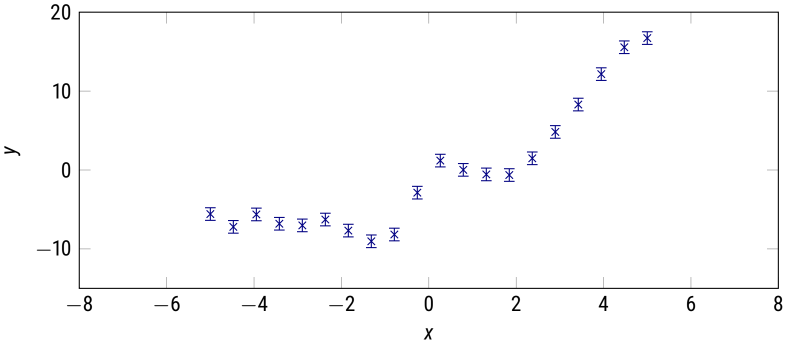
Bayesian inference becomes linear algebra

$$p(x) = \mathcal{N}(x; \mu, \Sigma) \qquad p(y \mid x) = \mathcal{N}(y; A^\intercal x + b, \Lambda)$$

$$p(B^\intercal x + c \mid y) = \mathcal{N}[B^\intercal x + c; B^\intercal \mu + c + B^\intercal \Sigma A(A^\intercal \Sigma A + \Lambda)^{-1}(y - A^\intercal \mu - b),$$
$$B^\intercal \Sigma B - B^\intercal \Sigma A(A^\intercal \Sigma A + \Lambda)^{-1} A^\intercal \Sigma B]$$

A Minimal Machine Learning Setup

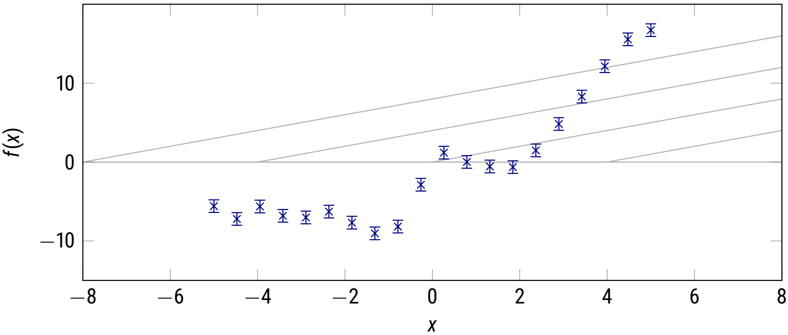
nonlinear regression problem



$$p(y \mid f_X) = \mathcal{N}(y; f_X, \sigma I)$$

Gaussian Parametric Regression

aka. general linear least-squares

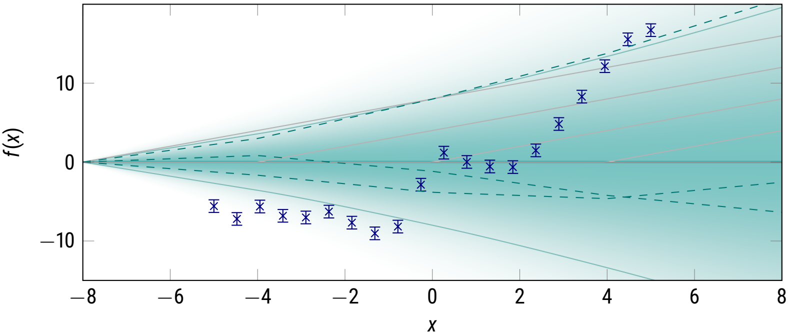


$$f(x) = \phi(x)^\intercal \mathbf{w} = \sum_i w_i \phi_i(x) \qquad p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mu, \Sigma)$$

$$\Rightarrow \quad p(f) = \mathcal{N}(f, \phi^\intercal \mu, \phi^\intercal \Sigma \phi) \qquad \phi_i(x) = \mathbb{I}(x > a_i) \cdot c_i (x - a_i) \qquad \text{(RELU)}$$

Gaussian Parametric Regression

aka. general linear least-squares

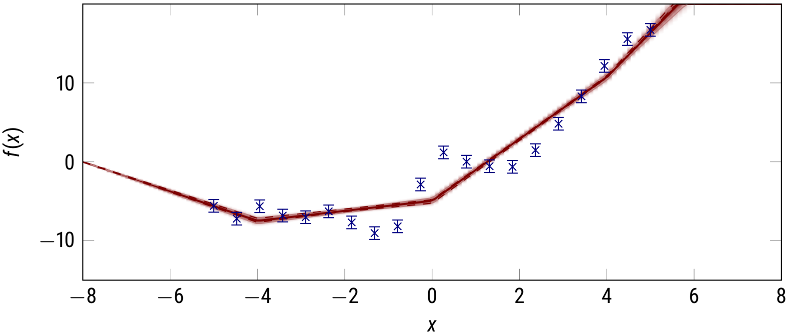


$$f(x) = \phi(x)^\intercal \mathbf{w} = \sum_i w_i \phi_i(x) \qquad p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mu, \Sigma)$$

$$\Rightarrow \quad p(f) = \mathcal{N}(f, \phi^\intercal \mu, \phi^\intercal \Sigma \phi) \qquad \phi_i(x) = \mathbb{I}(x > a_i) \cdot c_i(x - a_i) \qquad \text{(RELU)}$$

Gaussian Parametric Regression

aka. general linear least-squares

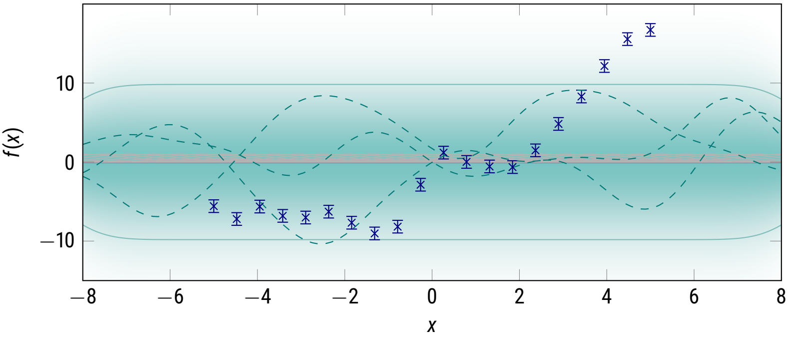


$$p(y \mid w, \phi_X) = \mathcal{N}(y; \phi_X^\intercal w, \sigma^2 I)$$

$$\begin{aligned} p(f_x \mid y, \phi_X) = \mathcal{N}(f_x; &\, \phi_x^\intercal \mu + \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1} (y - \phi_X^\intercal \mu), \\ &\, \phi_x^\intercal \Sigma \phi_x - \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1} \phi_X^\intercal \Sigma \phi_x) \end{aligned}$$

The Choice of Prior Matters

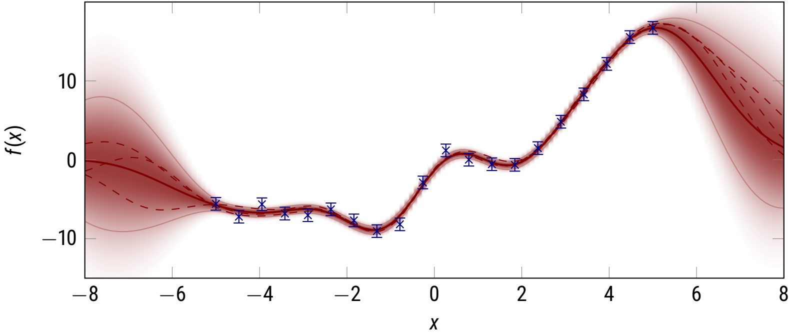
Bayesian framework provides flexible yet explicit modelling language



$$\phi_i(x) = \theta \exp\left(-\frac{(x - c_i)^2}{2\lambda^2}\right)$$

The Choice of Prior Matters

Bayesian framework provides flexible yet explicit modelling language



$$\phi_i(x) = \theta \exp\left(-\frac{(x - c_i)^2}{2\lambda^2}\right)$$

popular extension no. 1
requires large-scale linear algebra

$$p(f_x \mid y, \phi_X) = \mathcal{N}(f_x; \phi_x^\intercal \mu + \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1} (y - \phi_X^\intercal \mu),$$
$$\phi_x^\intercal \Sigma \phi_x - \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1} \phi_X^\intercal \Sigma \phi_x)$$

- set $\mu = \mathbf{0}$
- aim for closed-form expression of **kernel** $\phi_a^\intercal \Sigma \phi_b$

Features are cheap, so let's use a lot

an example

[DJC MacKay, 1998]

- For simplicity, let's fix $\Sigma = \frac{\sigma^2(c_{\max} - c_{\min})}{F} I$

$$\text{thus:} \quad \phi(x_i)^\intercal \Sigma \phi(x_j) = \frac{\sigma^2(c_{\max} - c_{\min})}{F} \sum_{\ell=1}^{F} \phi_\ell(x_i)\phi_\ell(x_j)$$

- especially, for $\phi_\ell(x) = \exp\left(-\frac{(x - c_\ell)^2}{2\lambda^2}\right)$

$$\begin{aligned}
&\phi(x_i)^\intercal \Sigma \phi(x_j) \\
&= \frac{\sigma^2(c_{\max} - c_{\min})}{F} \sum_{\ell=1}^{F} \exp\left(-\frac{(x_i - c_\ell)^2}{2\lambda^2}\right) \exp\left(-\frac{(x_j - c_\ell)^2}{2\lambda^2}\right) \\
&= \frac{\sigma^2(c_{\max} - c_{\min})}{F} \exp\left(-\frac{(x_i - x_j)^2}{4\lambda^2}\right) \sum_{\ell}^{F} \exp\left(-\frac{(c_\ell - \frac{1}{2}(x_i + x_j))^2}{\lambda^2}\right)
\end{aligned}$$

Features are cheap, so let's use a lot

an example [DJC MacKay, 1998]

$$\phi(x_i)^\intercal \Sigma \phi(x_j) =$$

$$\frac{\sigma^2(c_{\max} - c_{\min})}{F} \exp\left(-\frac{(x_i - x_j)^2}{4\lambda^2}\right) \sum_{\ell}^{F} \exp\left(-\frac{(c_\ell - \frac{1}{2}(x_i + x_j))^2}{\lambda^2}\right)$$

- now increase F so # of features in δc approaches $\frac{F \cdot \delta c}{(c_{\max} - c_{\min})}$

$$\phi(x_i)^\intercal \Sigma \phi(x_j) \to$$

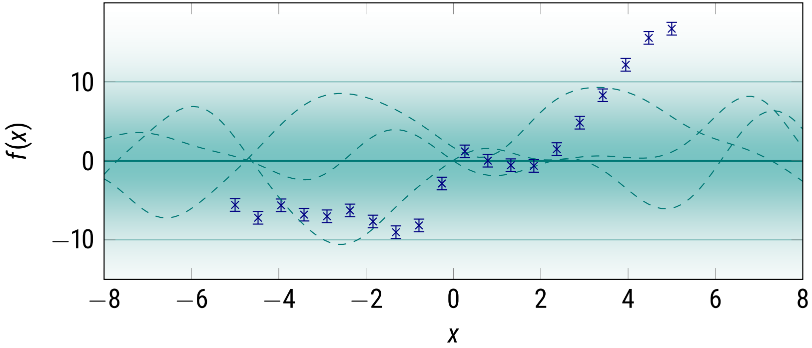
$$\sigma^2 \exp\left(-\frac{(x_i - x_j)^2}{4\lambda^2}\right) \int_{c_{\min}}^{c_{\max}} \exp\left(-\frac{(c - \frac{1}{2}(x_i + x_j))^2}{\lambda^2}\right) dc$$

- let $c_{\min} \to -\infty$, $c_{\max} \to \infty$

$$k(x_i, x_j) := \phi(x_i)^\intercal \Sigma \phi(x_j) \to \sqrt{2\pi}\lambda\sigma^2 \exp\left(-\frac{(x_i - x_j)^2}{4\lambda^2}\right)$$

Gaussian Process Regression

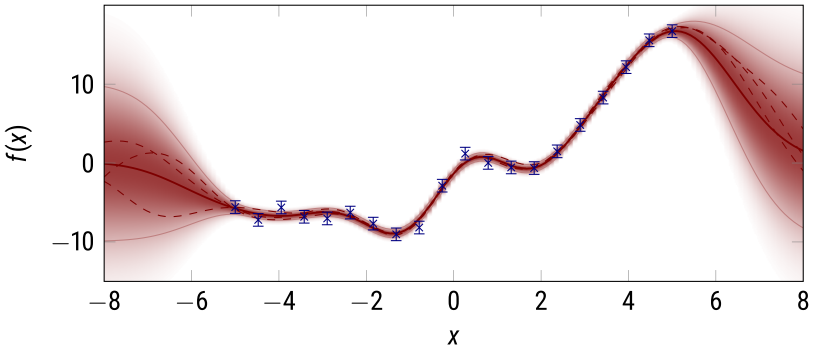
aka. Kriging, kernel-ridge regression,. . .



$$p(f) = \mathcal{GP}(0, k) \qquad k(a,b) = \exp\left(-\frac{(a-b)^2}{2\lambda^2}\right)$$

Gaussian Process Regression

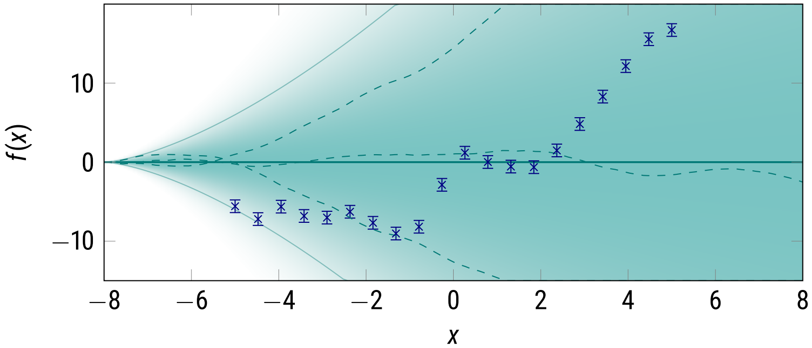
aka. Kriging, kernel-ridge regression,...



$$p(f \mid y) = \mathcal{GP}(f_x; k_{xX}(k_{XX} + \sigma^2 I)^{-1} y, k_{xx} - k_{xX}(k_{XX} + \sigma^2 I)^{-1} k_{Xx})$$

The prior still matters

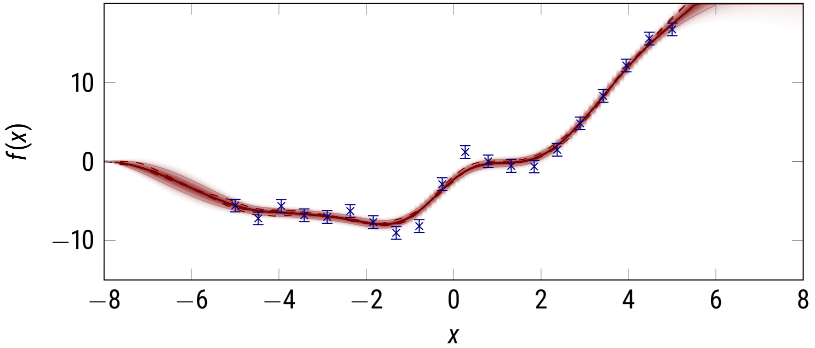
just one other example out of the space of kernels



For $\phi_i(x) = \mathbb{I}(x > c_i)(x - c_i)$, an analogous limit gives

The prior still matters

just one other example out of the space of kernels



$p(f) = \mathcal{GP}(0, k)$ with $k(a, b) = \theta^2 1/3 \min(a, b)^3 + |a - b| \min(a, b)^2$.
the **integrated Wiener process**, aka. **cubic splines**.

More on GPs in **Paris Perdikaris**' tutorial.

more on nonparametric models in **Neil Lawrence**'s and **Tamara Broderick**'s talks?

The Computational Challenge

large-scale linear algebra

$$\alpha := \underbrace{(k_{XX} + \sigma^2 I)^{-1}}_{\in \mathbb{R}^{N\times N}\text{, symm. pos. def.}} y \qquad k_{aX}(k_{XX} + \sigma^2 I)^{-1}k_{Xb} \qquad \log |k_{XX} + \sigma^2 I|$$

The Computational Challenge

large-scale linear algebra

$$\alpha := \underbrace{(k_{XX} + \sigma^2 I)^{-1}}_{\in \mathbb{R}^{N \times N}, \text{ symm. pos. def.}} y \qquad k_{aX}(k_{XX} + \sigma^2 I)^{-1} k_{Xb} \qquad \log |k_{XX} + \sigma^2 I|$$

Methods in wide use:

- exact linear algebra (BLAS), for $N \lesssim 10^4$ (because $\mathcal{O}(N^3)$)
- (rarely:) iterative Krylov solvers (in part. conjugate gradients), for $N \lesssim 10^5$

For large-scale ($\mathcal{O}(NM^2)$):

- inducing point methods, Nyström, etc.: **using iid. structure of data**

$$k_{ab} \approx \tilde{k}_{au} \Omega^{-1} \tilde{k}_{ub} \qquad \Omega^{-1} \in \mathbb{R}^{M \times M}$$

Williams & Seeger, 2001; Quiñonero & Rasmussen, 2005; Snelson & Ghahramani, 2007; Titsias, 2009

- spectral expansions **using algebraic properties of kernel**

Rahimi & Recht 2008; 2009

- in univariate setting: filtering **using Markov structure**

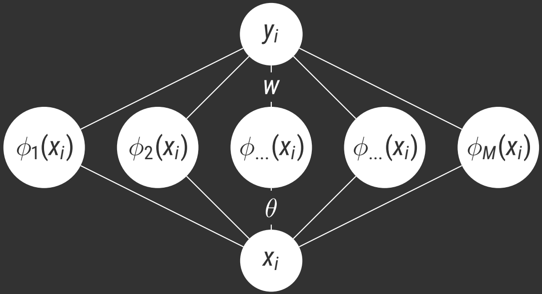
Särkkä 2013

Both are **linear time**, with **finite error**. Bridge to iterative methods is beginning to form, via **sub-space** recycling (de Roos & P.H., arXiv 1706.00241 2017)

popular extensions no. 2: requires large-scale nonlinear optimization

Maximum Likelihood estimation: Assume $\phi(x) = \phi_\theta(x)$

$$L(y;\theta,w) = \log p(y \mid \phi, w) = \frac{1}{2\sigma^2}\sum_{i=1}^{N} \|y_i - \phi_\theta(x_i)^\intercal w\|^2 + \text{const.}$$



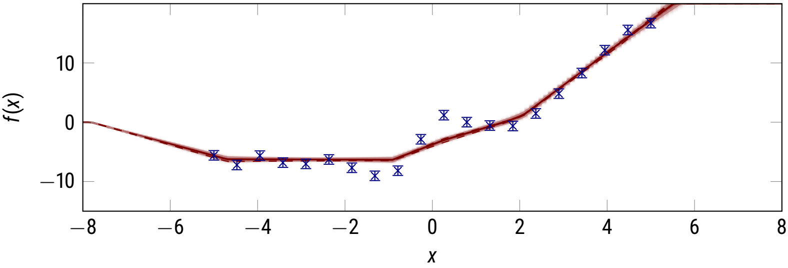
(A **feed-forward** network)

Learning Features

a (in general) **non-convex**, non-linear optimization problem

$$L(y;\theta,w) = \log p(y \mid \phi, w) = \frac{1}{2\sigma^2}\sum_{i=1}^{N} \|y_i - \phi_\theta(x_i)^\intercal w\|^2 + \text{const.}$$

$$\nabla_\theta L = \frac{1}{\sigma^2}\underbrace{\sum_{i=1}^{N} -(y_i - \phi_\theta(x_i)^\intercal w) \cdot w^\intercal \nabla_\theta \phi(x_i)}_{\text{“back-propagation”}}$$

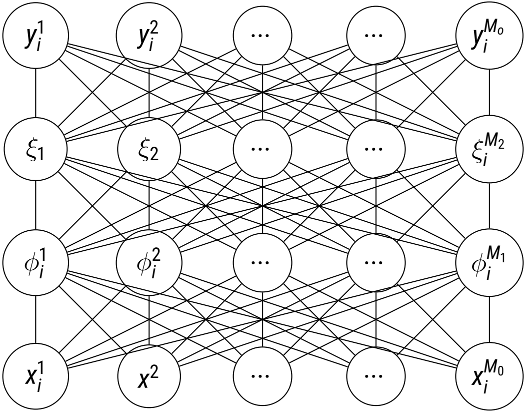


Deep Learning

(really just a quick peek)

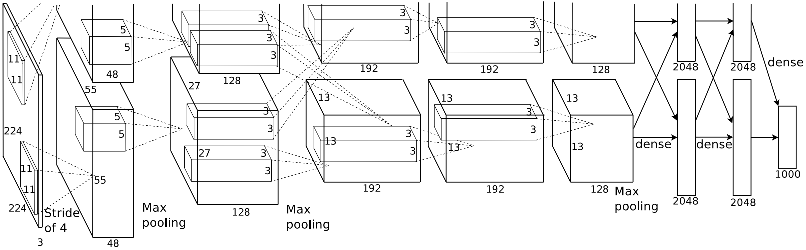
in practice:

- multiple input dimensions (e.g. pixel intensities)
- multi-dimensional output (e.g. structured sentences)
- multiple feature layers
- structured layers (convolutions, pooling, pyramids, etc.)



Deep Learning has become Mainstream

an increasingly professional industry



Krizhevsky, Sutskever & Hinton
“ImageNet Classification with Deep Convolutional Neural Networks”

Adv. in Neural Information Processing Systems (NIPS 2012) **25**, pp. 1097–1105

. . . and continues to impress

predicting whole-image semantic labels

man in black shirt is playing guitar. | construction worker in orange safety vest is working on road. | two young girls are playing with lego toy. | boy is doing backflip on wakeboard.

Karpathy & Fei-Fei."Deep Visual-Semantic Alignments for Generating Image Descriptions". *Computer Vision and Pattern Recognition* (CVPR 2015)

Zhao, Mathieu & LeCun, "Energy-based generative adversarial networks".
Int. Conf. on Learning Representations (ICLR) 2017

The Computational Challenge

high-dimensional, non-convex, **stochastic** optimization

- contemporary problems are extremely high-dimensional $N > 10^7$
- typically badly conditioned 🗎 Chaudhari et al. arXiv 1611.01838
- optimizer interacts with model
  🗎 Chaudhari et al. arXiv 1611.01838, 🗎 Keskar et al., 1609.04836
- biggest challenge: stochasticity

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i;\boldsymbol{\theta}) \approx \frac{1}{M}\sum_{j=1}^{M} \ell(y_j;\boldsymbol{\theta}) =: \hat{\mathcal{L}}(\boldsymbol{\theta}) \qquad M \ll N$$

$$p(\hat{\mathcal{L}} \mid \mathcal{L}) \approx \mathcal{N}\left(\hat{\mathcal{L}}; \mathcal{L}, \mathcal{O}\left(\frac{N-M}{M}\right)\right)$$

classic optimization paradigms break down.

- currently dominant optimizers are surprisingly simple:
  - stochastic gradient descent — Robbins & Monro, 1951
  - RMSPROP — Tielemann & Hinton, unpublished
  - ADADELTA — Zeiler, arXiv 1212.5701
  - ADAM — Kingma & Ba, ICLR 2015

more in part II ...

popular extension no. 3 requires high-dimensional integration of probability measures

- in $p(f) = \mathcal{GP}(0, k)$, what should k be?
- parametrize $k = k^\theta, \mu = \mu^\theta, \Lambda = \Lambda^\theta$

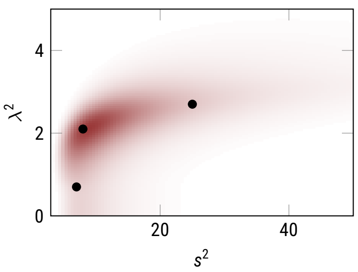
$$p(y \mid \theta) = \int p(y \mid f, \theta) p(f \mid \theta)\, df = \int \mathcal{N}(y; f_X, \Lambda^\theta) \mathcal{GP}(f; \mu^\theta, k^\theta)$$

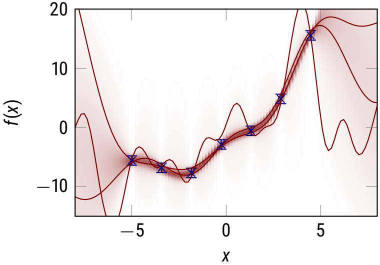
$$= \mathcal{N}(y, \mu_X^\theta, \Lambda^\theta + k_{XX}^\theta)$$

$$p(f \mid y) = \int p(f \mid y, \theta) p(\theta \mid y)\, d\theta$$

Learning the kernel

hierarchical Bayesian inference



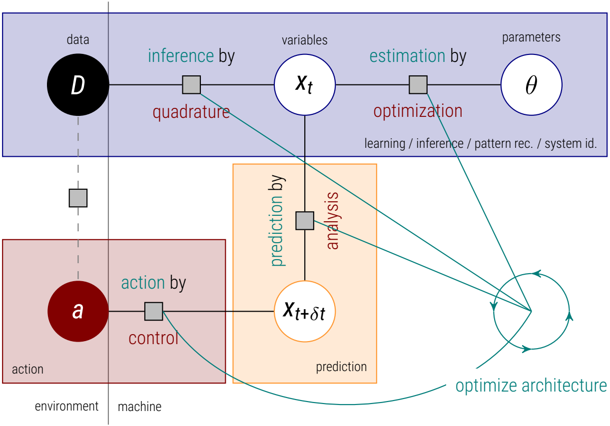


- practical cases can be extremely high-dimensional (⇾ Bayesian deep learning)
- standard approaches:
  - free energy minimization of a parametric approximation
  - Markov Chain Monte Carlo
- elaborate toolboxes available (⇾ probabilistic programming)
- but few (practically relevant) finite-time guarantees

more about hierarchical Bayesian inference in **Tamara Broderick**'s talk?

The Optimization View on Hierarchical Inference

Bayesian Optimization



- non-convex (multi-modal!) global optimization
- expensive evaluations

more about **optimization** of architectures in **Roman Garnett**'s talk

Summary: The Computations of Acting Agents

- machine intelligence requires computations
  - **integration** for marginalization
  - **optimization** for fitting
  - **differential equations** for control
  - **linear algebra** for all of the above
- contemporary AI problems pose very challenging numerical problems
- **uncertainty from data-subsampling** plays a crucial, intricate role
- classic numerical methods leave room for improvement

after coffee:
Learning machines don’t just pose problems—they also promise some answers.

Is there room at the bottom?

ML computations are dominated by numerical tasks

task …	…amounts to …	…using black box
marginalize	integration	MCMC, Variational, EP, …
train/fit	optimization	SGD et al., quasi-Nwton, …
predict/control	ord. diff. Eq.	Euler, Runge-Kutta, …
Gauss/kernel/LSq.	linear Algebra	Chol., CG, spectral, low-rank,…

- Scientific computing has produced a **very efficient toolchain**, but we are (usually) only using generic methods!
- **methods on loan** do not address some of ML's special needs
  - overly generic algorithms are inefficient
  - Big Data-specific challenges not addressed by "classic" methods

ML deservers customized numerical methods.
And as it turns out, we already have the right concepts!

Computation is Inference

http://probnum.org

Poincaré 1896, Kimeldorf & Wahba 1970, Diaconis 1988, O'Hagan 1992, ...

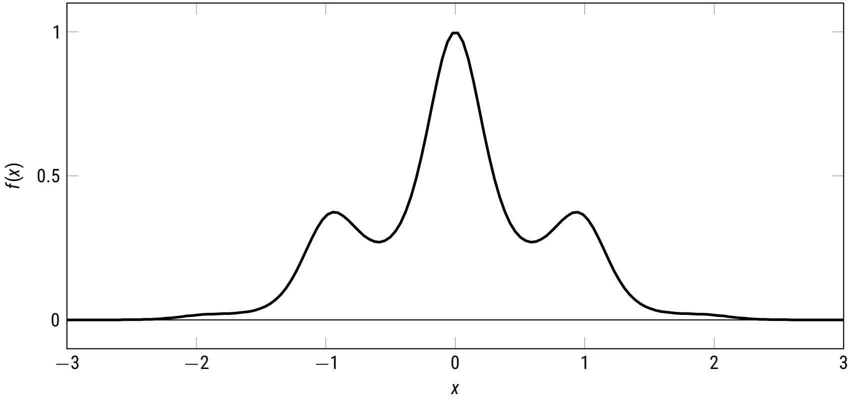
Numerical methods **estimate latent** quantities **given** the result of computations.

integration	estimate $\int_a^b f(x)\,dx$	given $\{f(x_i)\}$
linear algebra	estimate x s.t. $Ax = b$	given $\{As = y\}$
optimization	estimate x s.t. $\nabla f(x) = 0$	given $\{\nabla f(x_i)\}$
analysis	estimate $x(t)$ s.t. $x' = f(x, t)$	given $\{f(x_i, t_i)\}$

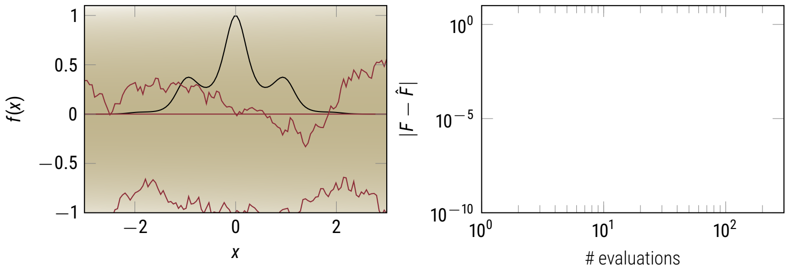
It is thus possible to build
probabilistic numerical methods
that use **probability measures** as in- and outputs,
and assign a notion of **uncertainty** to computation.

Integration

as Gaussian regression



$$f(x) = \exp(-\sin(3x)^2 - x^2) \qquad F = \int_{-3}^{3} f(x)\, dx = ?$$



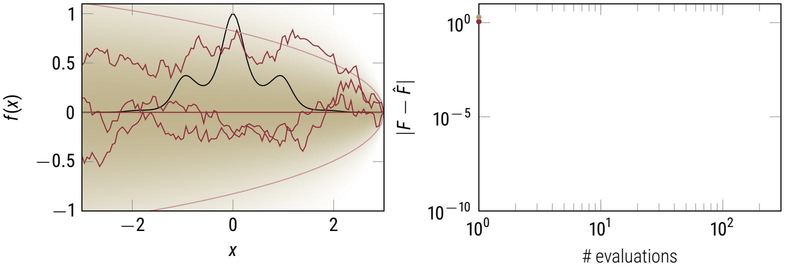
$$p(f) = \mathcal{GP}(f; 0, k) \qquad k(x, x') = \min(x, x') + c$$

$$\Rightarrow \; p\left(\int_a^b f(x)\,dx\right) = \mathcal{N}\left[\int_a^b f(x)\,dx; \int_a^b m(x)\,dx, \iint_a^b k(x, x')\,dx\,dx'\right]$$

$$= \mathcal{N}(F; 0, -{}^1\!/_6(b^3 - a^3) + {}^1\!/_2[b^3 - 2a^2 b + a^3] - (b - a)^2 c)$$

. . . conditioned on actively collected information . . .

computation as the collection of information

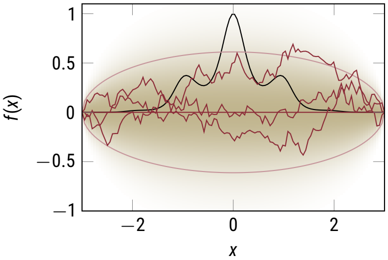


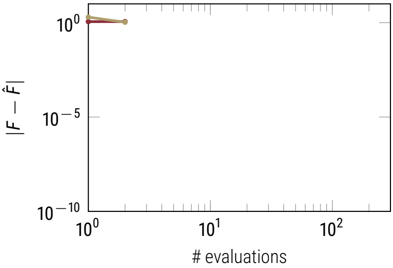
$$x_t = \arg\min \left[\mathrm{var}_{p(F|x_1,\dots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

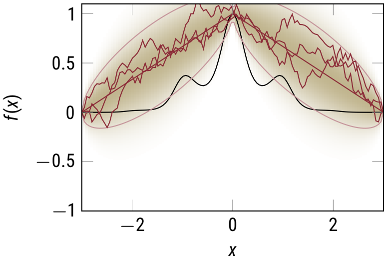
computation as the collection of information

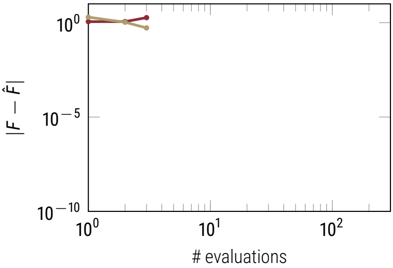




$$x_t = \arg\min \left[\operatorname{var}_{p(F|x_1,\dots,x_{t-1})}(F)\right]$$

- maximal reduction of variance yields regular grid



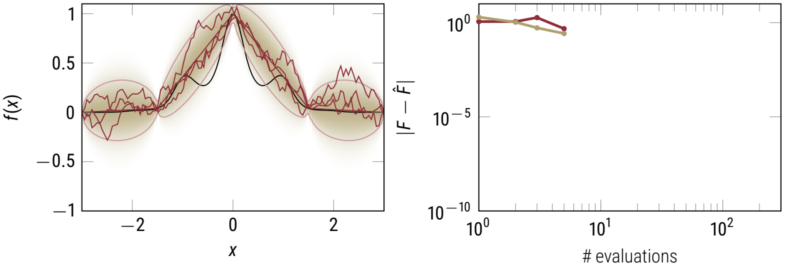


$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

computation as the collection of information

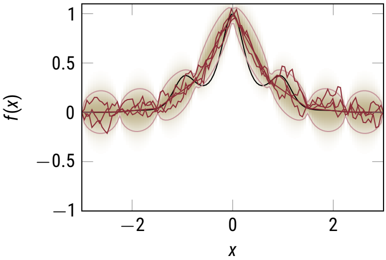


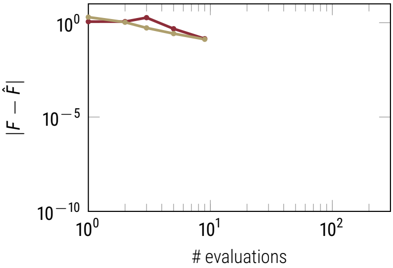
$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

... conditioned on actively collected information ...

computation as the collection of information



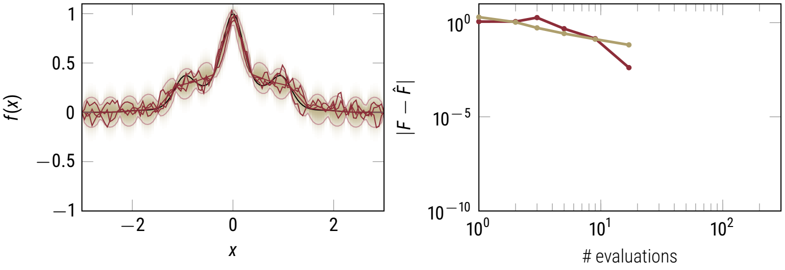


$$x_t = \arg\min \left[\operatorname{var}_{p(F|x_1,\dots,x_{t-1})}(F)\right]$$

- maximal reduction of variance yields regular grid

… conditioned on actively collected information …

computation as the collection of information

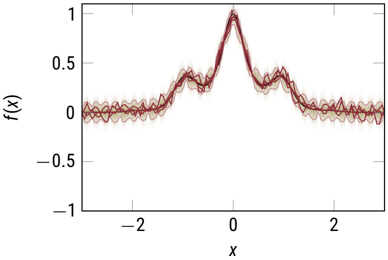


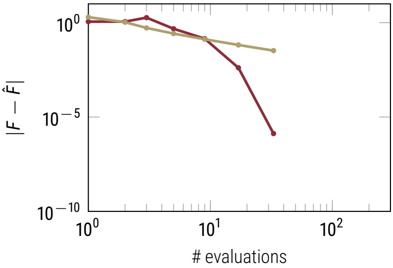
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

…conditioned on actively collected information …

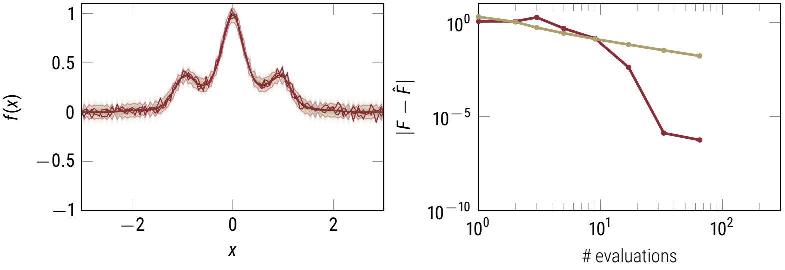
computation as the collection of information





$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

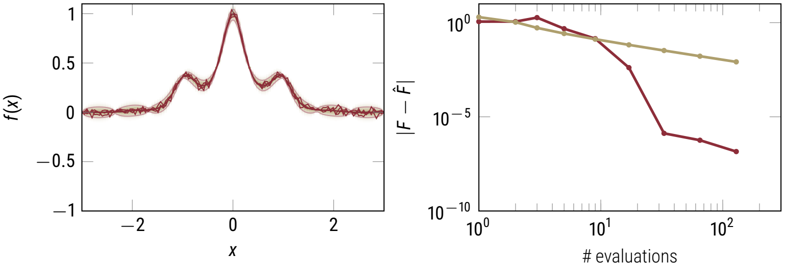


$$x_t = \arg\min \left[\operatorname{var}_{p(F|x_1,\ldots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

…conditioned on actively collected information …

computation as the collection of information

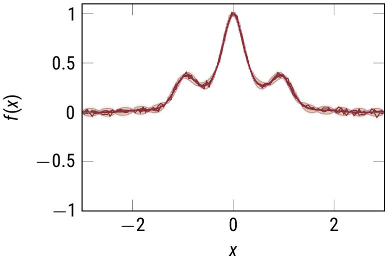


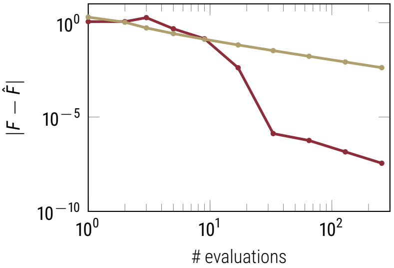
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

... conditioned on actively collected information ...

computation as the collection of information



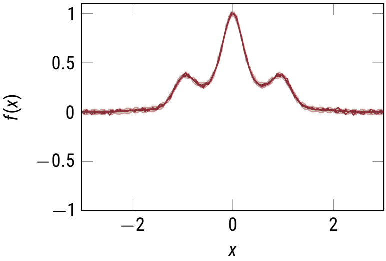


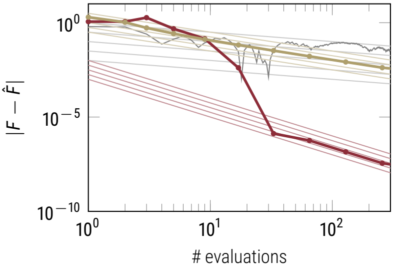
$$x_t = \arg\min \left[\mathrm{var}_{p(F|x_1,\dots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

computation as the collection of information



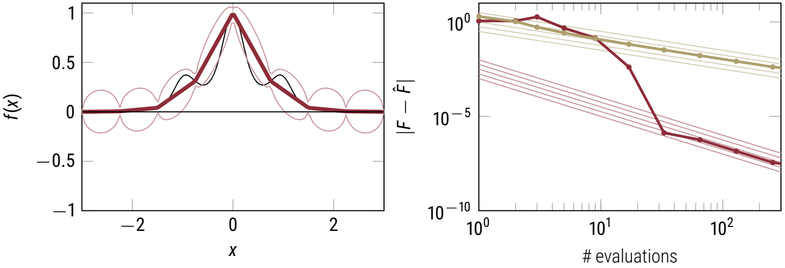


$$x_t = \arg\min \left[\mathrm{var}_{p(F|x_1,\dots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

. . . yields the trapezoid rule!

Kimeldorf & Wahba 1975, Diaconis 1988, O'Hagan 1985/1991



$$E_{\mathbf{y}}[F] = \int E_{|\mathbf{y}}[f(x)]\, dx = \sum_{i=1}^{N-1} (x_{i+1} - x_i) \frac{1}{2} (f(x_{i+1}) + f(x_i))$$

- **Trapezoid rule** is **MAP** estimate under Wiener process prior on f
- regular grid is optimal expected information choice
- error estimate is **under-confident**

more about **calibration** of uncertainty in the talks of **Chris Oates** and **John Cockayne**.

Computation as Inference

Bayes' theorem yields four levers for new functionality

Estimate ***z*** from computations ***c***, under model ***m***.

Prior:

Likelihood:

$$p(z \mid c, m) = \frac{p(z \mid m)p(c \mid z, m)}{\int p(z \mid m)p(c \mid z, m)\,dz}$$

Posterior:

Evidence:

Classic methods as basic probabilistic inference

maximum a-posteriori estimation in Gaussian models

[Ajne & Dalenius 1960; Kimeldorf & Wahba 1975; Diaconis 1988; O'Hagan 1985/1991]

Quadrature

Gaussian Quadrature $\longleftrightarrow$ GP Regression

[Hennig 2014]

Linear Algebra

Conjugate Gradients $\longleftrightarrow$ Gaussian Regression

[Hennig & Kiefel 2013]

Nonlinear Optimization

BFGS / Quasi-Newton $\longleftrightarrow$ Autoregressive Filtering

[Schober, Duvenaud & Hennig 2014; Kersting & Hennig 2016; Schober & Hennig 2016]

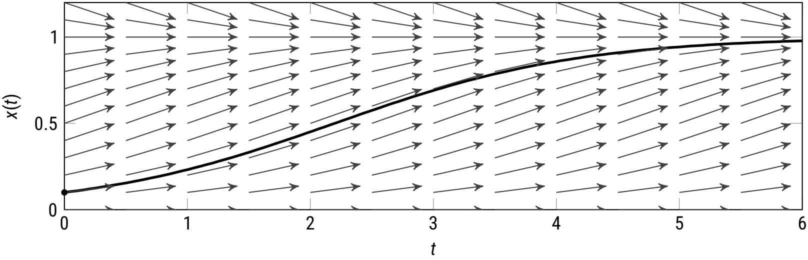
Differential Equations

Runge-Kutta; Nordsieck Methods $\longleftrightarrow$ Gauss-Markov Filters

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, ...

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)

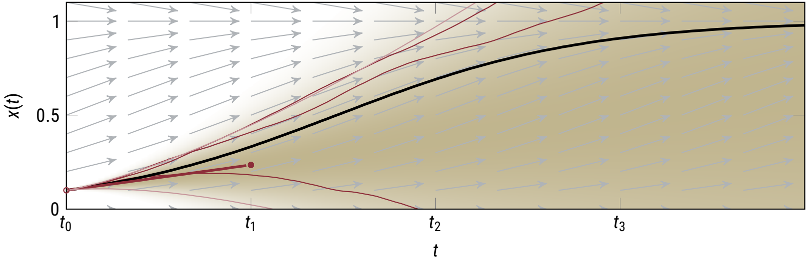
- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, ...

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)

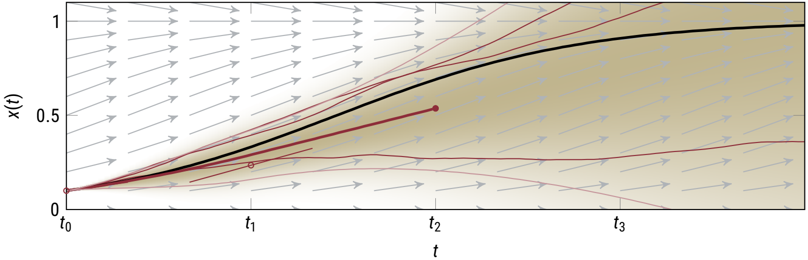
- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, . . .

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)

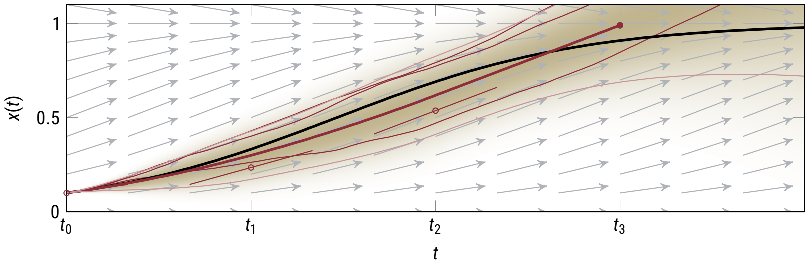
- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, ...

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + {}^1/_2$)

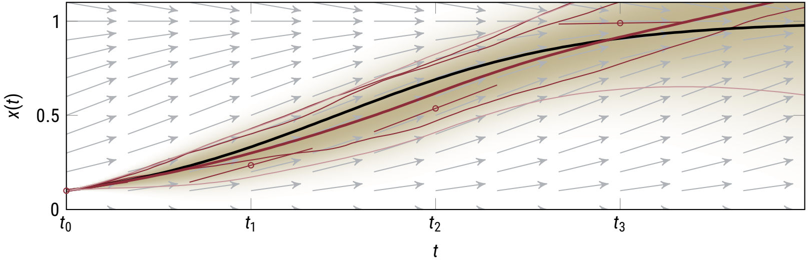
- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, ...

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)

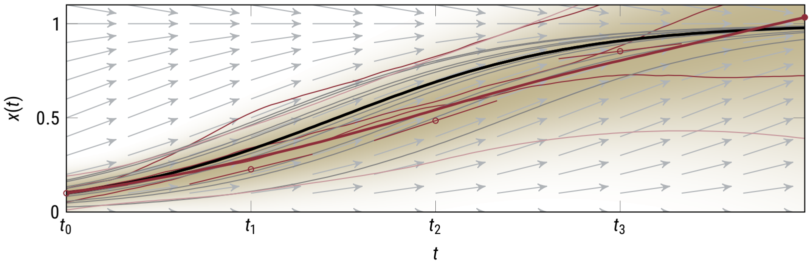
- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, ...

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)

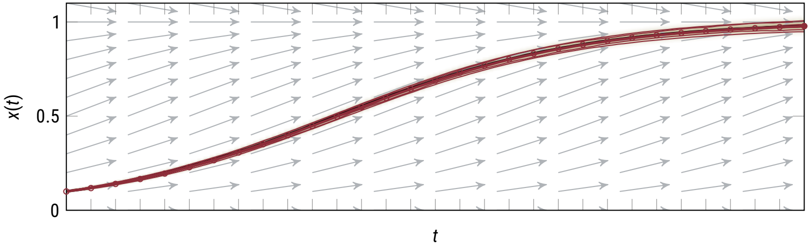
- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

Probabilistic ODE Solvers

Schober, Duvenaud & P.H., 2014. Schober & P.H., 2016. Kersting & P.H., 2016, ...

$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



There is a class of **solvers for initial value problems** that

- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)

- **this method** → **Hans Kersting**'s talk.
- **calibration** → **Oksana Chkrebtii**'s talk.
- **convergence** → **Tim Sullivan**'s talk.

https://github.com/ProbabilisticNumerics/pfos

- Probabilistic numerics can be as **fast** and **reliable** as classic ones.
- **Computation can be phrased on ML language!**
- Meaningful (**calibrated**) uncertainty can be constructed at minimal computational overhead (dominated by cost of point estimate)

So what does this mean for Data Science / ML / AI?

New Functionality, and new Challenges

making use of the probabilistic numerics perspective

Prior: structural knowledge reduces complexity.

Likelihood:

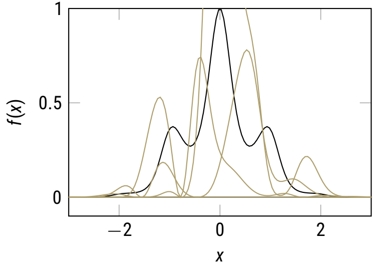
$$p(z \mid c, m) = \frac{p(z \mid m)p(c \mid z, m)}{\int p(z \mid m)p(c \mid z, m)\, dz}$$

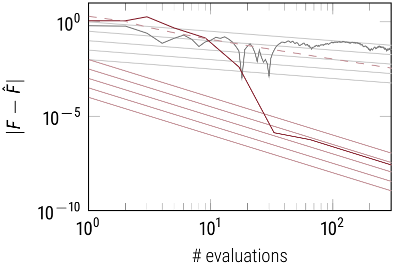
Posterior:

Evidence:

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🗎 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014





a prior specifically for integration of probability measures

- $f > 0$ (f is probability measure)
- $f \propto \exp(-x^2)$ (f is product of prior and likelihood terms)
- $f \in \mathcal{C}^\infty$ (f is smooth)

Explicit prior knowledge yields reduces complexity.

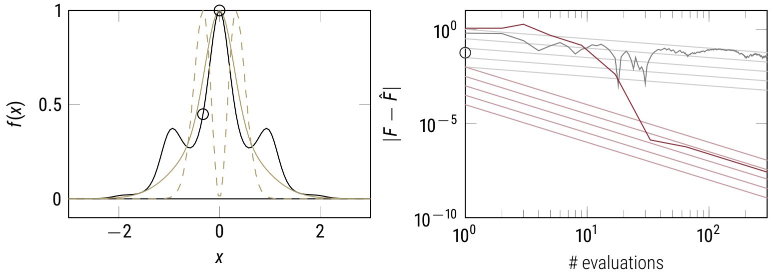
cf. **information-based complexity**.

e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🗎 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014



- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

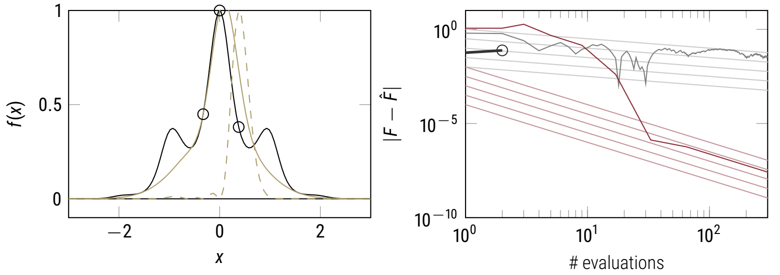
cf. **information-based complexity**.

e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🗎 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014



- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

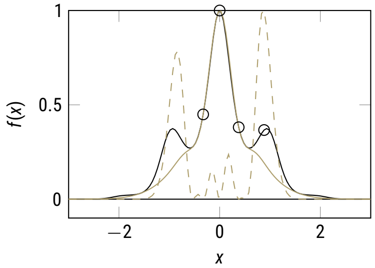
cf. **information-based complexity**.

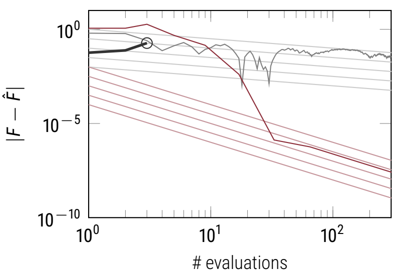
e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014





- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

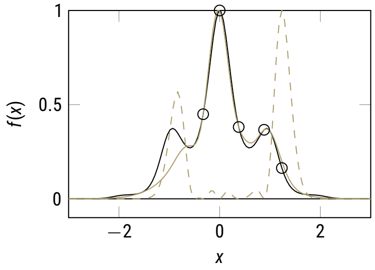
cf. **information-based complexity**.

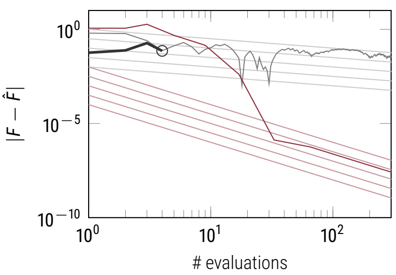
e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🗎 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014





- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

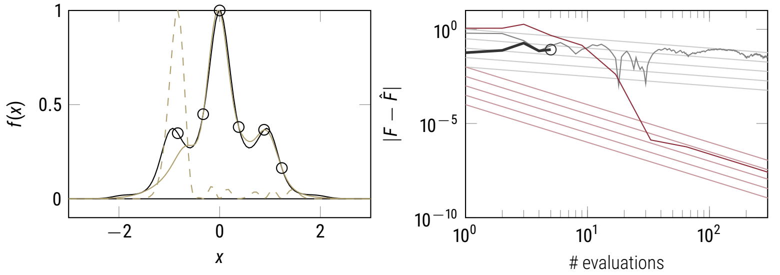
cf. **information-based complexity**.

e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🖹 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014



- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

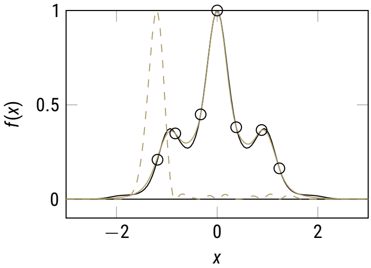
cf. **information-based complexity**.

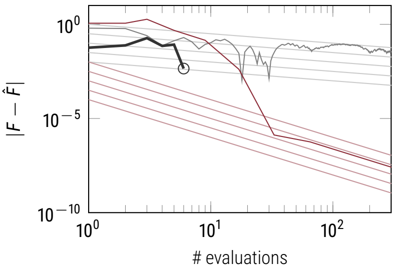
e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🗎 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014





- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

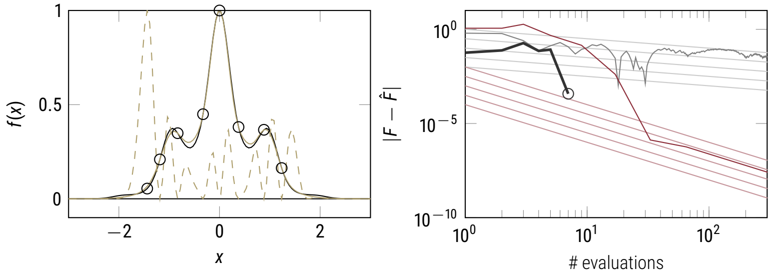
cf. **information-based complexity**.

e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) 🗎 Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014



- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

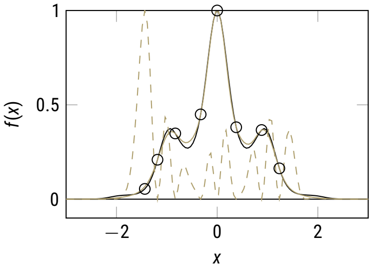
cf. **information-based complexity**.

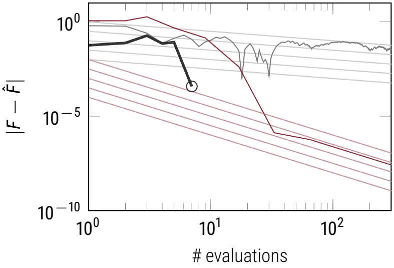
e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

An integration prior for probability measures

WArped Sequential Active Bayesian Integration (WSABI) Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014





- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

Explicit prior knowledge yields reduces complexity.

cf. **information-based complexity**.

e.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2

more on this connection in **Houman Owhadi**'s tutorial?

Computation as Inference

new numerical functionality for machine learning

Estimate z from computations c, under model m.

Prior: structural knowledge reduces complexity

Likelihood: modeling imprecise computation reduces cost

$$p(z \mid c, m) = \frac{p(z \mid m)\,p(c \mid z, m)}{\int p(z \mid m)\,p(c \mid z, m)\,dz}$$

Posterior:

Evidence:

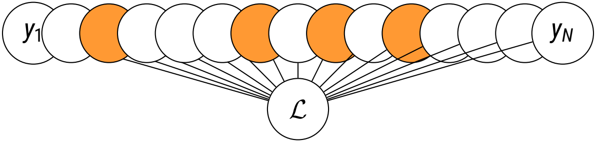
New numerics for Big Data

Uncertainty on Inputs directly effecting numerical decisions

In Big Data setting, batching introduces (Gaussian) noise

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\ell(y_i;\boldsymbol{\theta}) \approx \frac{1}{M}\sum_{j=1}^{M}\ell(y_j;\boldsymbol{\theta}) =: \hat{\mathcal{L}}(\boldsymbol{\theta}) \qquad M \ll N$$

$$p(\hat{\mathcal{L}} \mid \mathcal{L}) \approx \mathcal{N}\left(\hat{\mathcal{L}}; \mathcal{L}, \mathcal{O}\left(\frac{N-M}{M}\right)\right)$$



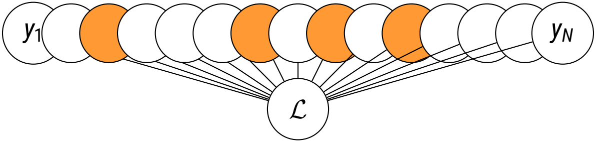
New numerics for Big Data

Uncertainty on Inputs directly effecting numerical decisions

In Big Data setting, batching introduces (Gaussian) noise

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\ell(y_i;\boldsymbol{\theta}) \approx \frac{1}{M}\sum_{j=1}^{M}\ell(y_j;\boldsymbol{\theta}) =: \hat{\mathcal{L}}(\boldsymbol{\theta}) \qquad M \ll N$$

$$p(\hat{\mathcal{L}} \mid \mathcal{L}) \approx \mathcal{N}\left(\hat{\mathcal{L}}; \mathcal{L}, \mathcal{O}\left(\frac{N-M}{M}\right)\right)$$



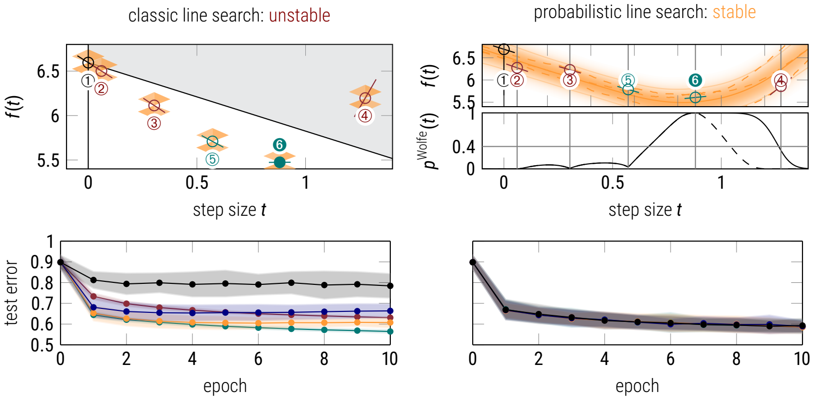
Classic methods are unstable to noise. E.g.: step size selection

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \nabla \hat{\mathcal{L}}(\theta_t)$$

Probabilistic Line Searches

Step-size selection stochastic optimization

Mahsereci & Hennig, NIPS 2015



two-layer feed-forward perceptron on CIFAR 10. Details, additional results in Mahsereci & Hennig, NIPS 2015.

https://github.com/ProbabilisticNumerics/probabilistic_line_search

- **batch-size selection** — cabs — Balles & Hennig, arXiv 1612.05086
- **early stopping** — Mahsereci, Balles & Hennig, arXiv 1703.09580
- **search directions** — sodas — Balles & Hennig, arXiv 1705.07774

Computation as Inference

new numerical functionality for machine learning

Estimate z from computations c, under model m.

Prior: structural knowledge reduces complexity

Likelihood: modeling imprecise computation reduces cost

$$p(z \mid c, m) = \frac{p(z \mid m)p(c \mid z, m)}{\int p(z \mid m)p(c \mid z, m)\, dz}$$

Posterior: tracking uncertainty for robustness

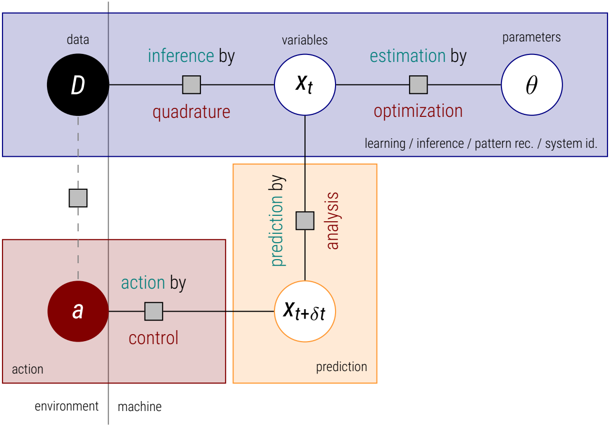
Evidence:

cf. Hennig, Osborne, Girolami, Proc. Royal Soc. A, 2015

Uncertainty Across Composite Computations

interacting information requirements

Hennig, Osborne, Girolami, Proc. Royal Society A 2015



- probabilistic numerical methods taking and producing uncertain inputs and outputs allow management of computational resources

more on uncertainty propagation in **Ilias Bilionis**' talk.

Computation as Inference

new numerical functionality for machine learning

Estimate ***z*** from computations ***c***, under model ***m***.

Prior: structural knowledge reduces complexity

Likelihood: modeling imprecise computation reduces cost

$$p(z \mid c, m) = \frac{p(z \mid m)\, p(c \mid z, m)}{\int p(z \mid m)\, p(c \mid z, m)\, dz}$$

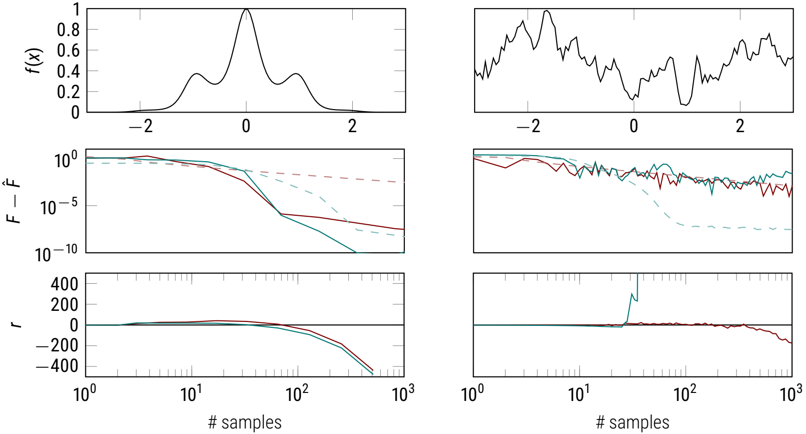
Posterior: tracking uncertainty for robustness

Evidence: checking models for safety

cf. Hennig, Osborne, Girolami, Proc. Royal Soc. A, 2015

Probabilistic Certification?

proof of concept: Hennig, Osborne, Girolami. Proc. Royal Society A, 2015



$$r = \mathrm{E}_{\tilde{f}}\left[\log \frac{p(\tilde{f}(\mathbf{x}))}{p(f(\mathbf{x}))}\right] = (f(\mathbf{x}) - \mu(\mathbf{x}))^\intercal K^{-1}(f(\mathbf{x}) - \mu(\mathbf{x})) - N$$

Summary

Uncertain computation **as** and **for** machine learning

- **computation is inference → probabilistic numerical methods**
  - probability measures for **uncertain** inputs and outputs
  - classic methods as special cases

New concepts not just for Machine Learning:

prior: structural knowledge reduces complexity
likelihood: imprecise computation lowers cost
posterior: uncertainty propagated through computations
evidence: model mismatch detectable at run-time

- ML & AI pose **new** computational challenges
- computational methods can be phrased in the concepts **of** ML
- **but** related results of mathematics are currently "under-explored"
- more about all of this in **this seminar**!

http://probnum.org **https://pn.is.tue.mpg.de**