Probabilistic Numerics

Uncertainty in Computation

Philipp Hennig

Stuttgart
24 September 2018

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

erc some of the presented work is supported by the European Research Council.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN
Max Planck Institute for Intelligent Systems

Machine Learning needs

Data — **Computations** — Model

integration	like	MCMC, VMP, EP, …	for	probabilistic inference
optimization		SGD, Adam, RMSprop, …		(stochastic) fitting
differential eqs.		Runge-Kutta, Multi-Step, …		forecasting & control
linear algebra		Cholesky, CG, spectral, …		all of the above

generic methods save design time, but do not address special needs

- overly generic algorithms are inefficient
- Big Data-specific challenges not addressed by “classical” methods

Data Science / AI / ML needs to build its own numerical methods.
As it turns out, we already have the right concepts!

Computation is Inference

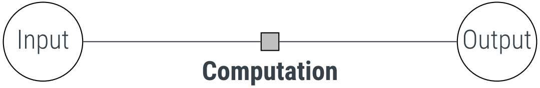
http://probnum.org

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

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

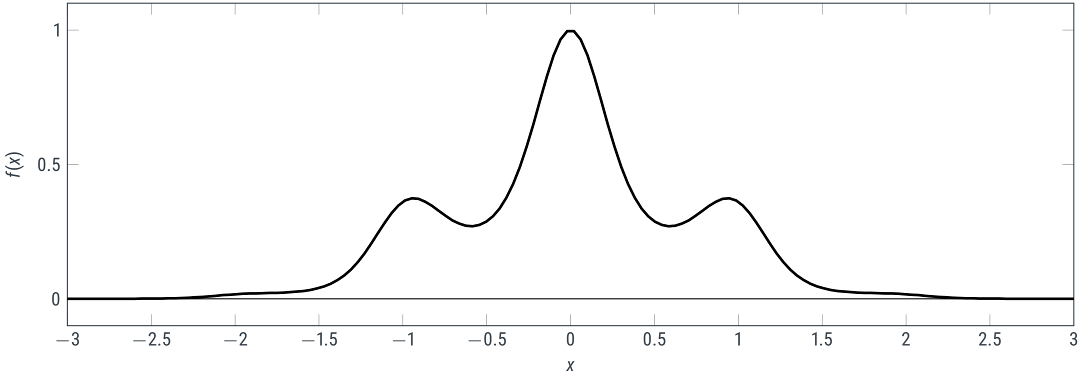
integration	estimate	$\int_a^b f(x)\,dx$	given $\{f(x_i)\}$
linear algebra	estimate	x s.t. $Ax = b$	given $\{As = y\}$
optimization	estimate	x s.t. $\nabla f(x) = 0$	given $\{\nabla f(x_i)\}$
simulation	estimate	$x(t)$ s.t. $x' = f(x, t)$	given $\{f(x_i, t_i)\}$

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

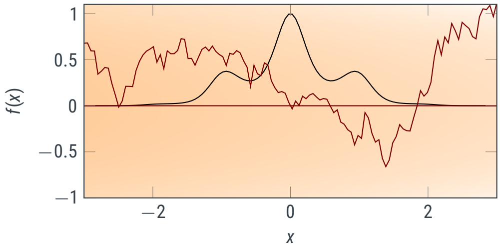


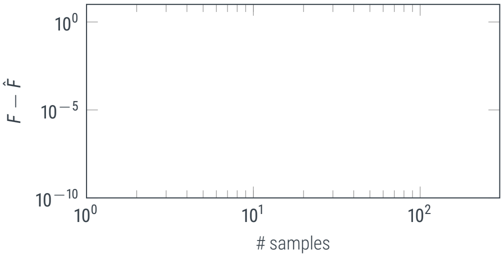
Integration

as Gaussian regression



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

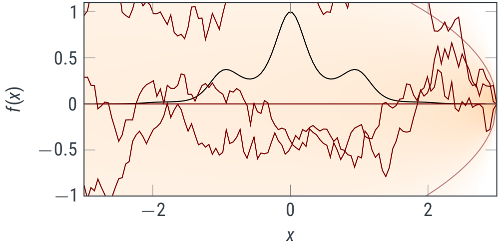


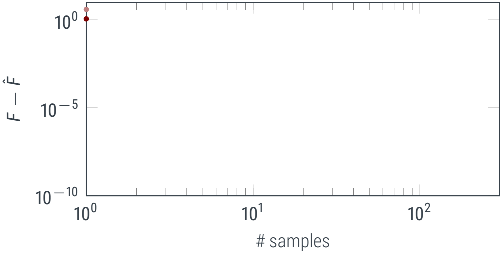


$$p(f) = \mathcal{GP}(f; 0, k) \qquad k(x, x') = \min(x, x') + c$$

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

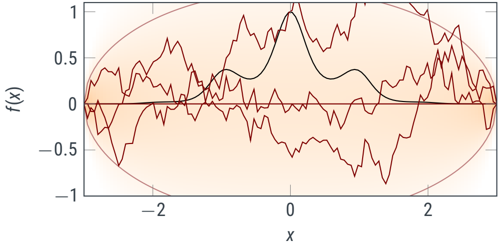
$$= \mathcal{N}(F; 0, -1/6(b^3 - a^3) + 1/2[b^3 - 2a^2b + a^3] - (b - a)^2 c)$$

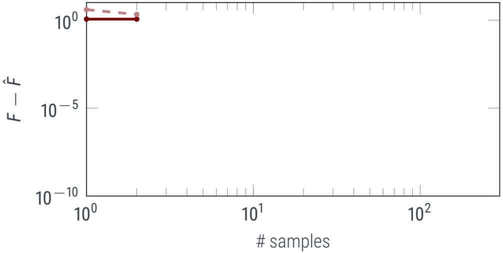




$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1,\dots,x_{t-1})}(F)\right]$$

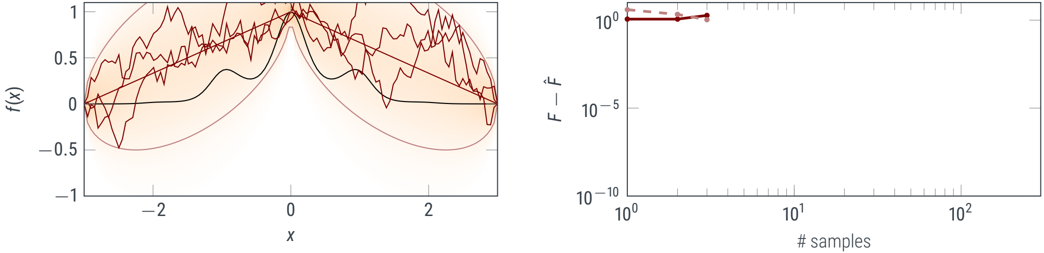
- **maximal reduction of variance** yields **regular grid**





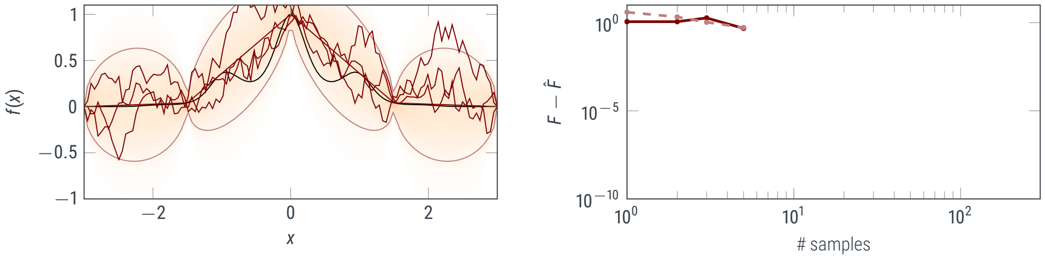
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

- **maximal reduction of variance** yields **regular grid**



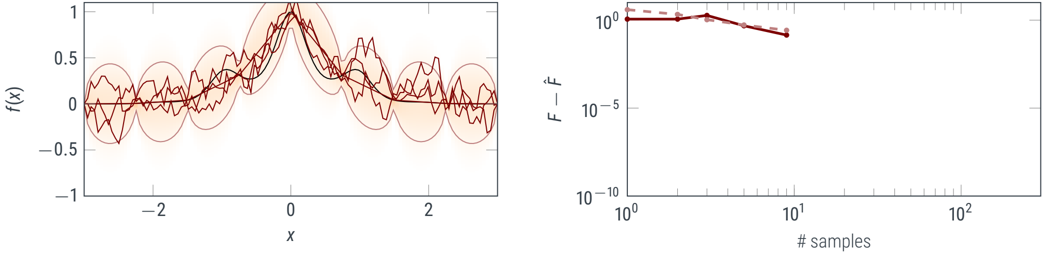
$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- **maximal reduction of variance** yields **regular grid**



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

- **maximal reduction of variance** yields **regular grid**

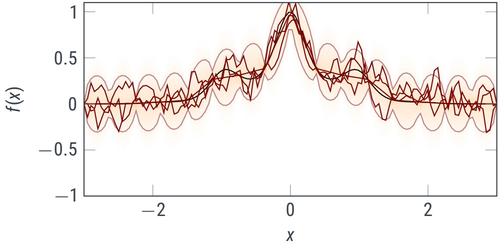


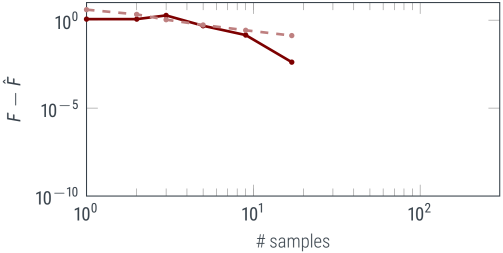
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

- **maximal reduction of variance** yields **regular grid**

…conditioned on actively collected information …

computation as the collection of information



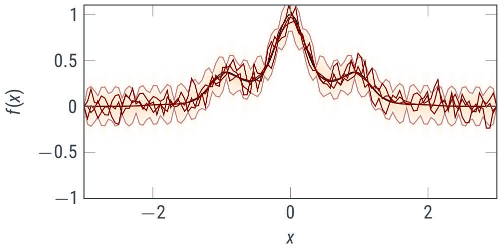


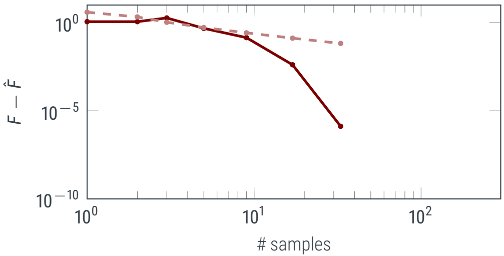
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1,\dots,x_{t-1})}(F) \right]$$

- **maximal reduction of variance** yields **regular grid**

. . . conditioned on actively collected information . . .

computation as the collection of information



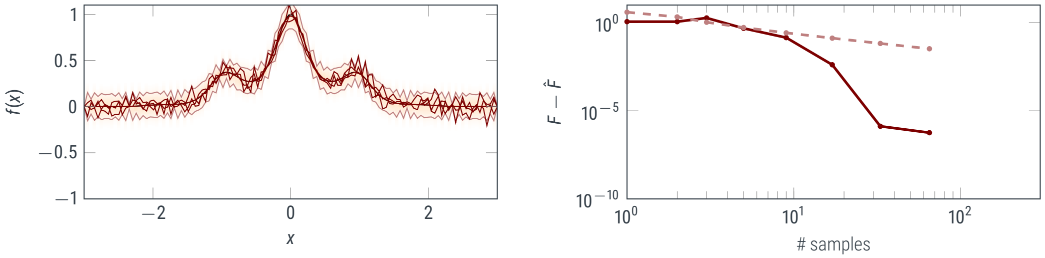


$$x_t = \arg\min \left[\mathrm{var}_{p(F|x_1,\dots,x_{t-1})}(F) \right]$$

- **maximal reduction of variance** yields **regular grid**

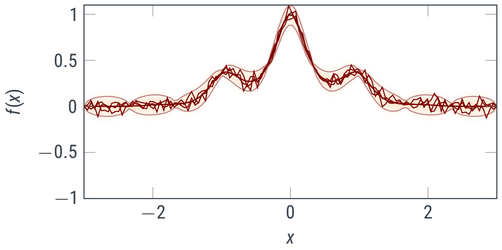
. . . conditioned on actively collected information . . .

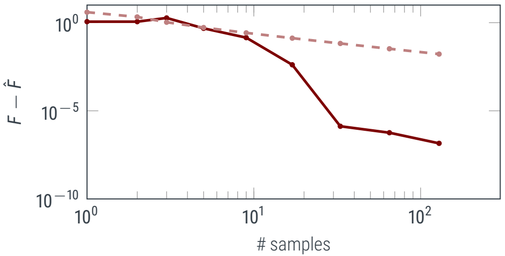
computation as the collection of information



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

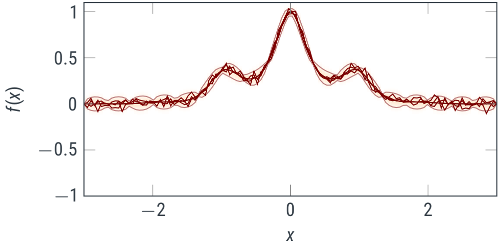
- **maximal reduction of variance** yields **regular grid**

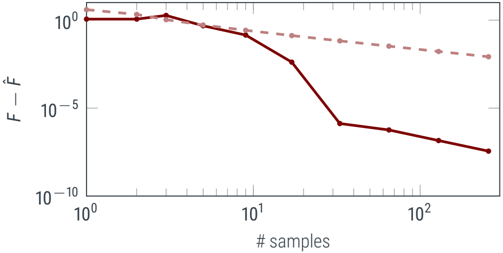




$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

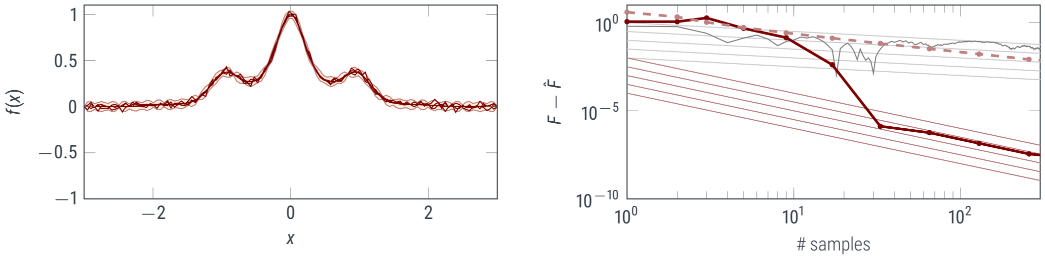
- **maximal reduction of variance** yields **regular grid**





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

- **maximal reduction of variance** yields **regular grid**



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

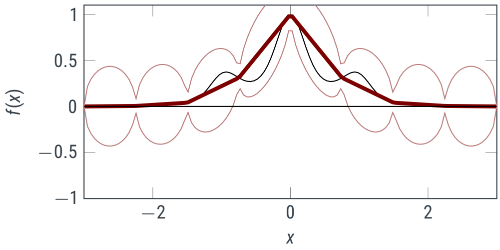
- **maximal reduction of variance** yields **regular grid**

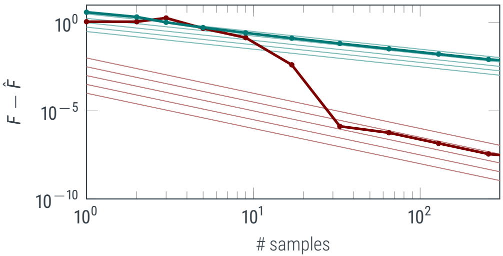
... yields the trapezoid rule!

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

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





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

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

Computation as Inference

Bayesian inference on a latent (non-analytic) quantity from computable "observations"

Estimate z from computations c, under model m.

$$\underbrace{p(z \mid c, m)}_{\text{posterior}} = \frac{\overbrace{p(z \mid m)}^{\text{prior}} \overbrace{p(c \mid z, m)}^{\text{likelihood}}}{\underbrace{\int p(z \mid m) p(c \mid z, m)\, dz}_{\text{evidence}}}$$

Classic methods as basic probabilistic inference

maximum a-posteriori estimation in Gaussian models

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

Gaussian Quadrature ⟷ GP Regression

Linear Algebra [Hennig 2014]

Conjugate Gradients ⟷ Gaussian Regression

Nonlinear Optimization [Hennig & Kiefel 2013]

BFGS / Quasi-Newton ⟷ Autoregressive Filtering

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

Runge-Kutta; Nordsieck Methods ⟷ Gauss-Markov Filters

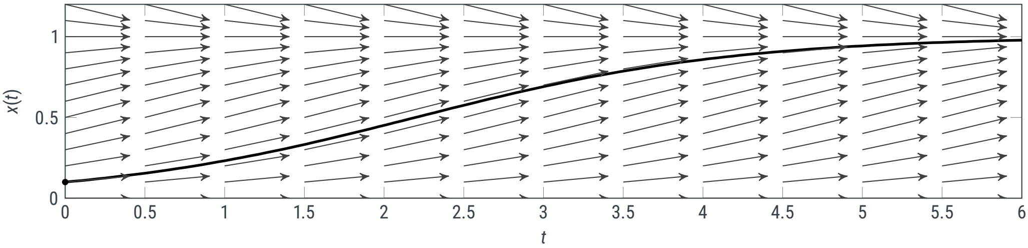
Probabilistic numerical methods can be as **fast** and **reliable** as classic ones.

Probabilistic ODE Solvers

Same story, different task

[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 $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

Probabilistic ODE Solvers

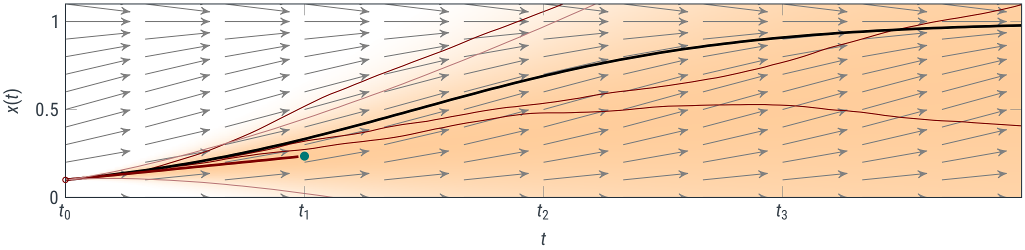
Same story, different task

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

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

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



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

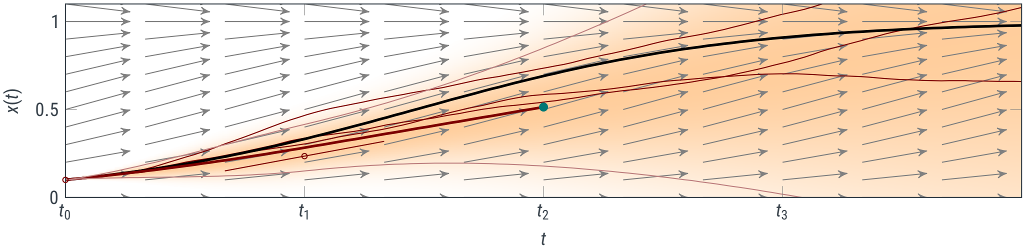
- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

Probabilistic ODE Solvers

Same story, different task

[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

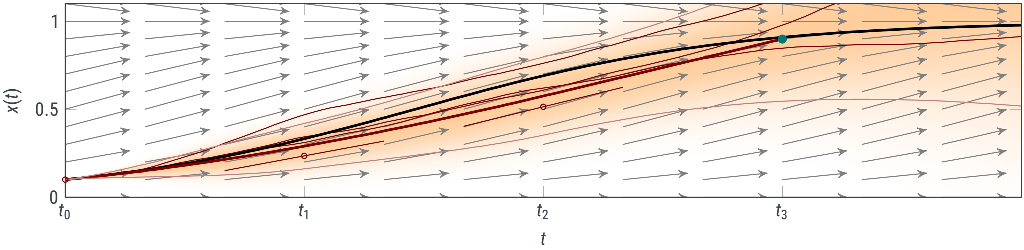
- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

Probabilistic ODE Solvers

Same story, different task

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

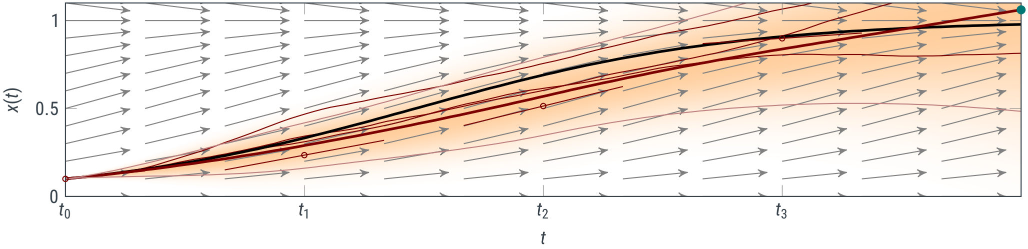
$$x'(t) = f(x(t), t), \quad x(t_0) = x_0$$



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 $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

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



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

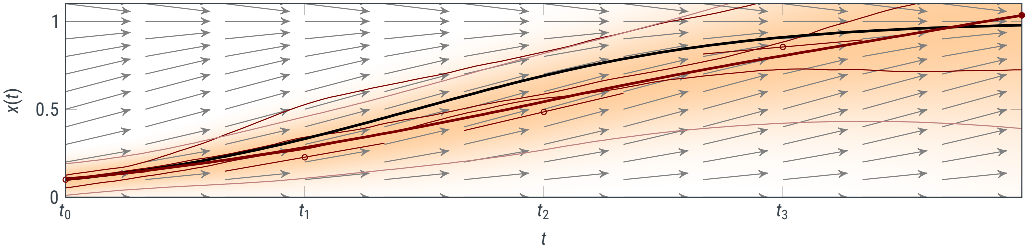
- has the same **complexity** as multi-step methods
- has **high local approximation order** q (like classic solvers)
- has **calibrated posterior uncertainty** (order $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

Probabilistic ODE Solvers

Same story, different task

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

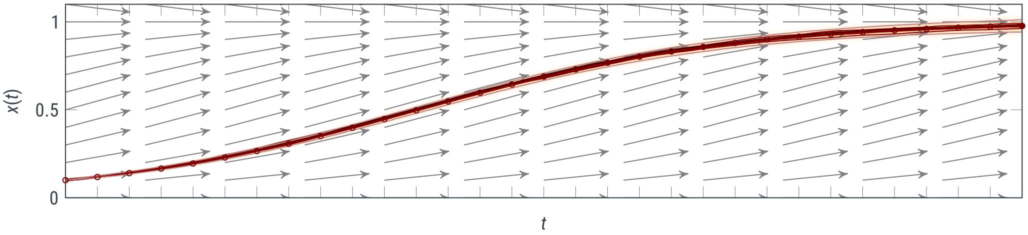
$$x'(t) = f(x(t), t), \qquad x(t_0) = x_0$$



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 $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

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



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 $q + 1/2$)
- can use **uncertain initial value** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

- Computation is an instance of **inference**.
- many classic numerical methods can be interpreted as probabilistic inference, arising from specific **generative models** (prior & likelihood)
- Meaningful (calibrated) uncertainty can be constructed at minimal computational overhead (dominated by cost of point estimate)
- **Designing a numerical method is a modelling task!**

The probabilistic viewpoint allows **new functionality** for **contemporary challenges**.

Prior: structural knowledge reduces complexity.

Likelihood:

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

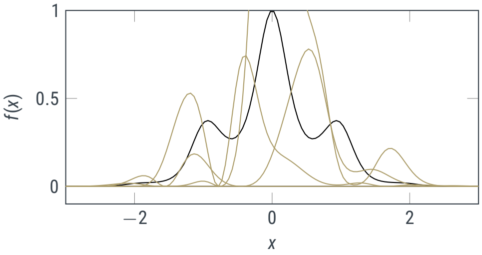
Posterior:

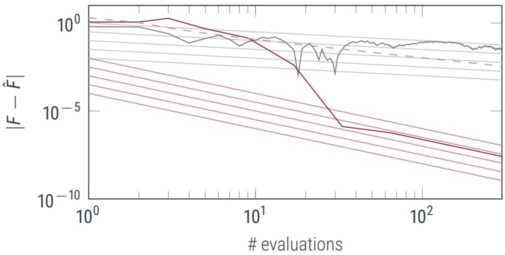
Evidence:

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)

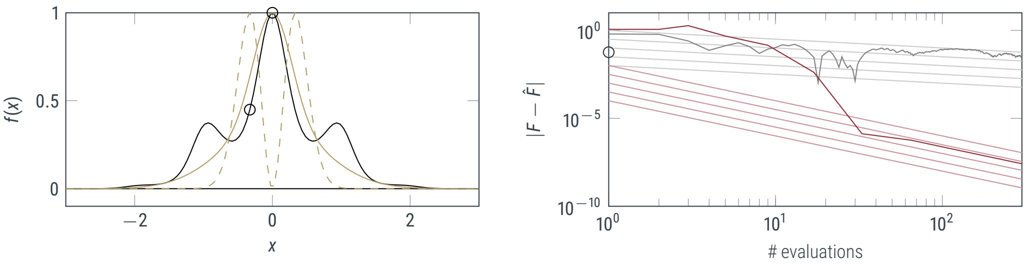
Explicit prior knowledge yields reduces complexity.

[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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

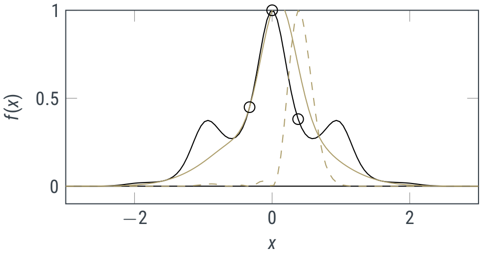
Explicit prior knowledge yields reduces complexity.

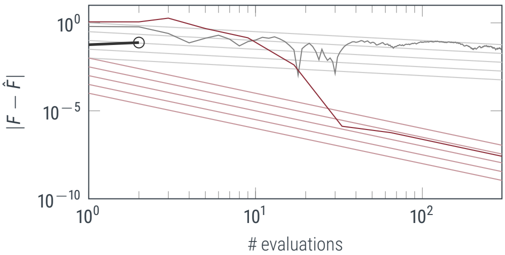
[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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

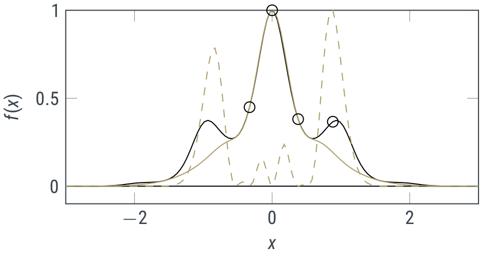
Explicit prior knowledge yields reduces complexity.

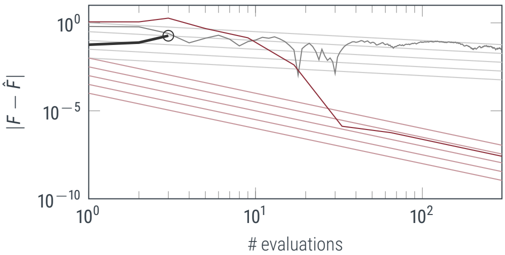
[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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

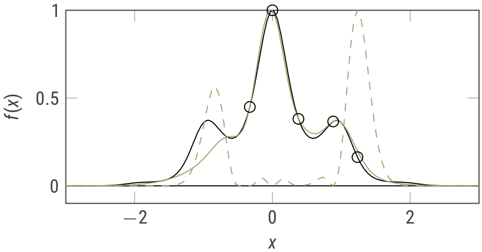
Explicit prior knowledge yields reduces complexity.

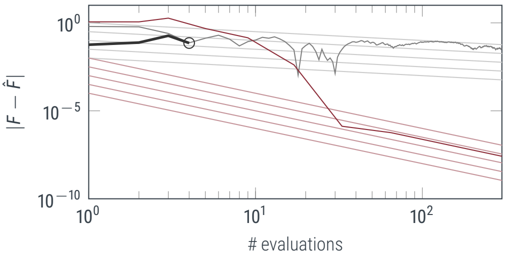
[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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]

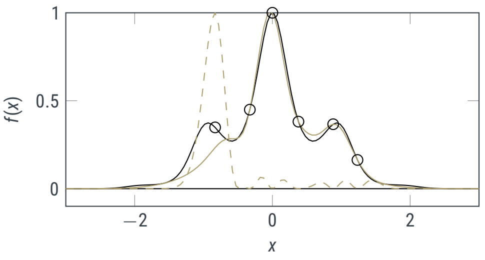
An integration prior for probability measures

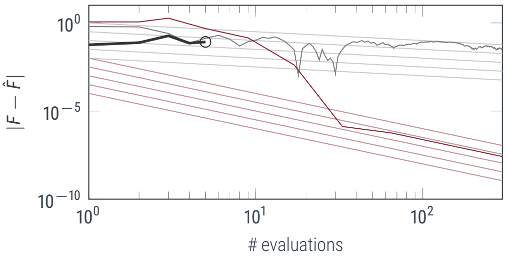
WArped Sequential Active Bayesian Integration (WSABI)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

[Gunter, Osborne, Garnett, Hennig, Roberts. NIPS 2014]





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

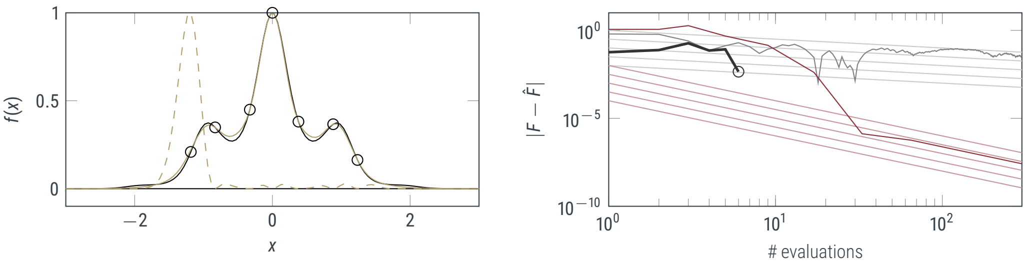
Explicit prior knowledge yields reduces complexity.

[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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

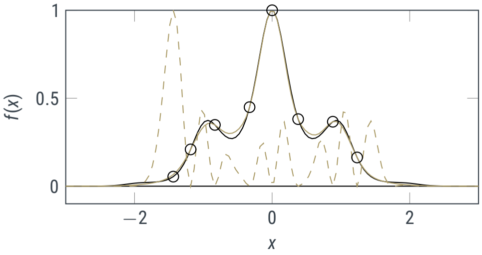
Explicit prior knowledge yields reduces complexity.

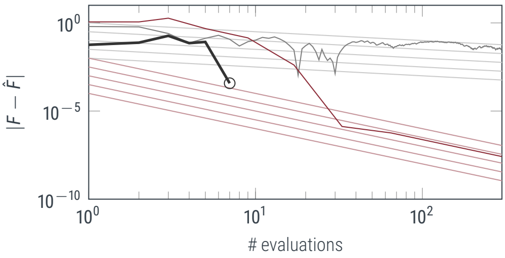
[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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

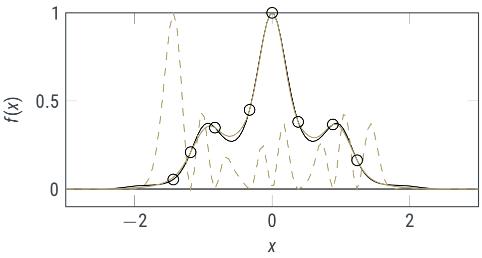
Explicit prior knowledge yields reduces complexity.

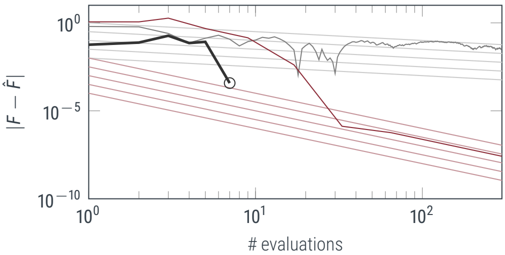
[cf. **information-based complexity**. E.g. Novak, 1988. Clancy et al. 2013, arXiv 1303.2412v2]

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]

Estimate z from computations c, under model m.

Prior: structural knowledge reduces complexity

Likelihood: modelling imprecision stabilizes algorithms

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

Posterior:

Evidence:

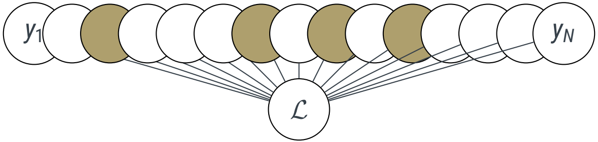
The usual assumption:

$$p(c \mid z, m) = \delta(c - A_m z)$$

In Big Data setting, iid. batching introduces Gaussian noise

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

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



In Big Data setting, iid. batching introduces Gaussian noise

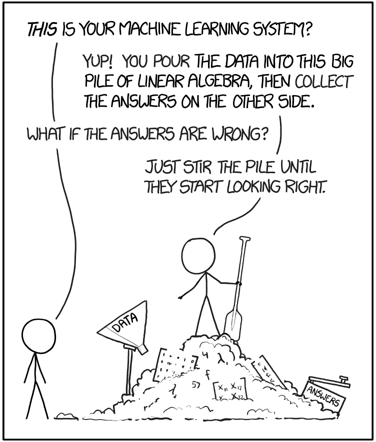
$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i; \boldsymbol{\theta}) \approx \frac{1}{M}\sum_{j=1}^{M} \ell(y_j; \boldsymbol{\theta}) =: \hat{\mathcal{L}}(\boldsymbol{\theta})$$

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

Contemporary machine learning requires tedious parameter fitting.

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

- step size / learning rate α_t
- batch size M
- number of steps to termination
- search directions

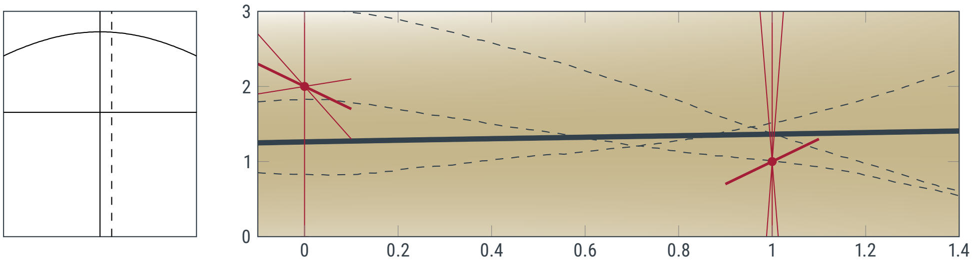


http://xkcd.com/1838

Uncertainty Can Induce Free Parameters

and require new observables to identify them

[Balles, Mahsereci, Hennig (ICML-AutoML 2017)]



$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i;\theta) \approx \frac{1}{M}\sum_{j=1}^{M} \ell(y_j;\theta) =: \hat{\mathcal{L}}(\theta) \qquad M \ll N \qquad p(\hat{\mathcal{L}} \mid \mathcal{L}) \approx \mathcal{N}\left(\hat{\mathcal{L}}; \mathcal{L}, \mathcal{O}\left(\frac{1}{M}\right)\right)$$

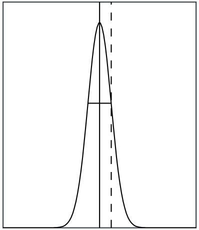
Uncertainty Can Induce Free Parameters

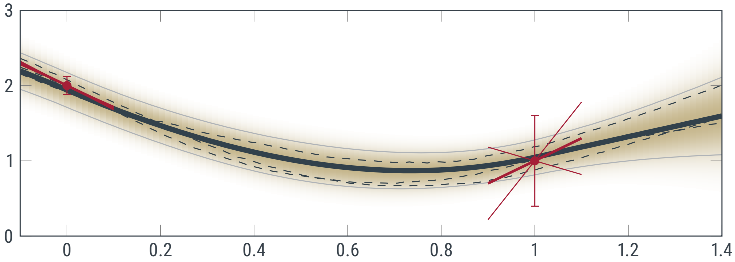
and require new observables to identify them

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

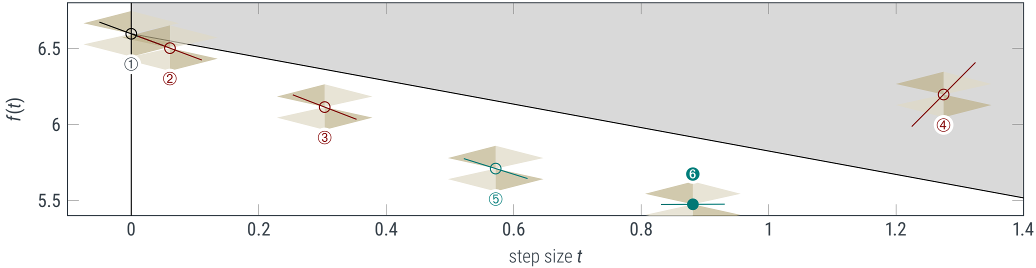
[Balles, Mahsereci, Hennig (ICML-AutoML 2017)]





$$\operatorname{var}\hat{\mathcal{L}}(\theta) \approx \frac{1}{M-1}\left(\frac{1}{M}\sum_{j=1}^{M}\ell^2(y_j;\theta) - \hat{L}^2(\theta)\right) \qquad p(\hat{\mathcal{L}} \mid \mathcal{L}) \approx \mathcal{N}\left(\hat{\mathcal{L}}; \mathcal{L}, \operatorname{var}\hat{\mathcal{L}}\right)$$

Capturing the likelihood requires a **new observable**! It's computation is not free, but cheap!
But without it, a key algorithmic parameter is **unidentified**!



- $f'(t_{\text{cand}}) > 0$? bisect : extend
- until **Wolfe conditions** are fulfilled:

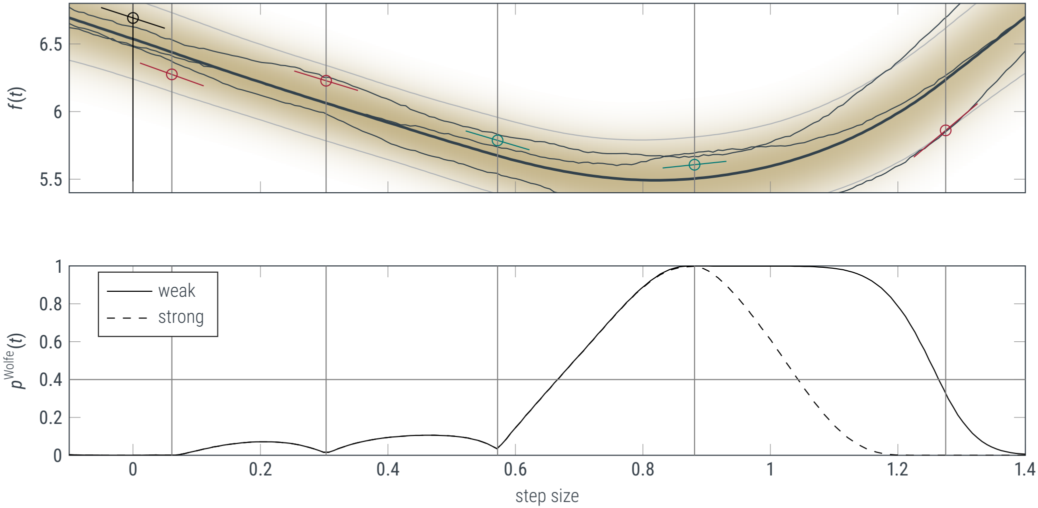
$$f(t) < f(0) + c_1 f'(0) \quad \text{AND} \quad |f'(t)| < c_2 |f'(0)|$$

Probabilistic Line Searches

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

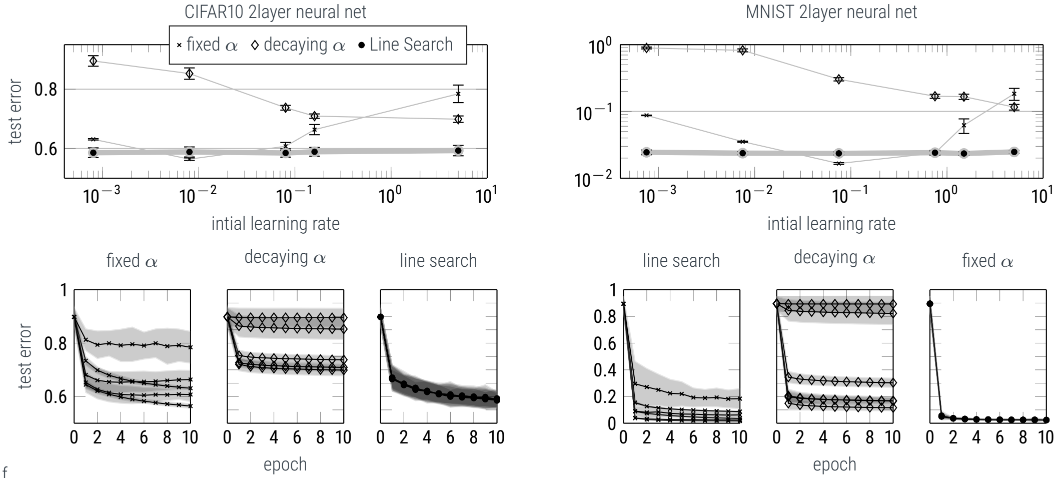
Max Planck Institute for Intelligent Systems

[Mahsereci & Hennig, NIPS 2015 (oral) / JMLR 2017]



No more Learning Rates!

two-layer feed-forward perceptron. Details, additional results: Mahsereci & Hennig, JMLR **18**(119):1–59, 2017.



f

https://github.com/ProbabilisticNumerics/probabilistic_line_search

Choosing Batch Sizes

trading off cost and precision

[Balles, Romero, Hennig, UAI 2017]

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

- trade-off: **std**$[\nabla\hat{\mathcal{L}}] = \mathcal{O}(1/\sqrt{M})$, but cost is $\mathcal{O}(M)$
- for SGD: lower bound on **improvement**: Assume $\nabla\mathcal{L}$ Lipschitz

$$\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1}) \geq G := \alpha\nabla\mathcal{L}(\theta_t)^\intercal\nabla\hat{\mathcal{L}}(\theta_t) - \frac{L\alpha^2}{2}\|\nabla\hat{\mathcal{L}}(\theta_t)\|^2$$

expected improvement: under $p(\hat{\mathcal{L}} \mid \mathcal{L})$ $\quad \mathbb{E}(G) = \left(\alpha - \frac{L\alpha^2}{2}\right)\|\nabla\mathcal{L}(\theta_t)\|^2 - \frac{L\alpha^2}{2M}\sum_\ell \mathrm{var}[\nabla_\ell\hat{\mathcal{L}}(\theta_t)]$

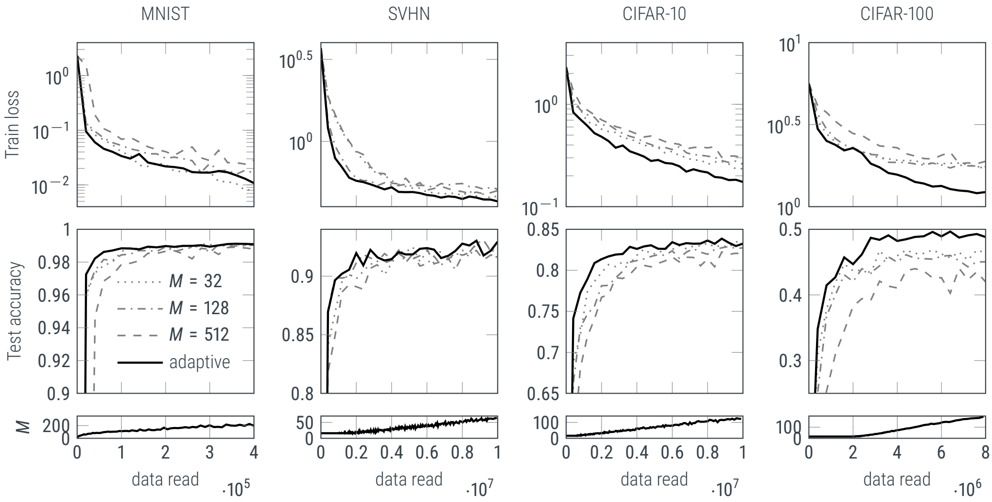
- maximize **expected improvement per cost**, let **line-search** find $\alpha = 1/L$, some further simplifications (local 2nd order approximation, assert **min** $\mathcal{L} \gtrsim 0$),

$$M_* = \arg\max_M \frac{\mathbb{E}[G]}{M} \approx \alpha_t \frac{\sum_\ell \mathrm{var}[\nabla_\ell\hat{\mathcal{L}}(\theta_t)]}{\hat{\mathcal{L}}(\theta_t)}$$

Choosing Batch-Sizes

trading off cost and precision

[Balles, Romero, Hennig, UAI 2017]



https://github.com/ProbabilisticNumerics/CABS

- in empirical risk minimization, just figuring out when to **stop** the optimizer is a non-trivial problem
- even the full data set is a sample relative to the population
- **overfitting** becomes a problem when gradients (with their estimatable variance) are statistically indistinguishable to white noise around zero

$$\log p(\nabla\hat{\mathcal{L}} \mid \nabla\mathcal{L} = 0) > \mathbb{E}_{p(\nabla\hat{\mathcal{L}}|\nabla\mathcal{L}=0)}\left[\log p(\nabla\hat{\mathcal{L}} \mid \nabla\mathcal{L} = 0)\right]$$

$$1 - \frac{M}{D}\sum_{\ell=1}^{D}\frac{(\nabla_\ell\mathcal{L}(\theta_t))^2}{\operatorname{var}\nabla_\ell\hat{\mathcal{L}}(\theta_t)} > 0 \quad \Rightarrow \quad \text{STOP!}$$

- **step sizes** — Mahsereci & Hennig
  Probabilistic Line Searches for Stochastic Optimization — NIPS 2015
  `https://github.com/ProbabilisticNumerics/probabilistic_line_search`
- **batch sizes** — Balles, Romero, Hennig
  Coupling Adaptive Batch Sizes with Learning Rates — UAI 2017
  `https://github.com/ProbabilisticNumerics/cabs`
- **termination criteria** — Mahsereci, Balles, Lassner, Hennig
  Early Stopping without a Validation Set — arXiv 1703.09580

- **data sub-sampling** gives rise to imprecise computations / non-Dirac observations **likelihoods**
- **free algorithmic parameters** may then become **un-identified**
- likelihood shape can be identified with **minor computational overhead**
- **classic methods** provide a **blue-print**
- re-phrasing them probabilistically allow **inference** on free parameters

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

Prior: structural knowledge reduces complexity

Likelihood: modelling imprecision stabilizes algorithms

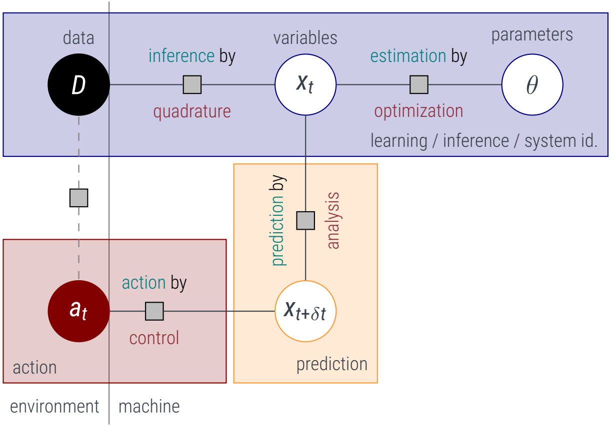
$$p(z \mid c, m) = \frac{p(z \mid m)p(c \mid z, m)}{\int p(z \mid m)p(c \mid z, m)\, dz}$$

Posterior: tracking uncertainty for robustness

Evidence:

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

[Hennig, Osborne, Girolami, Proc. Royal Soc. A, 2015]



for some recent theory, see Thm. 5.9 in Cockayne, Oates, Sullivan, Girolami. arXiv 1702.03673

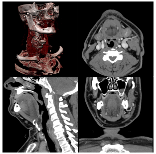
Probabilistic Treatment Planning

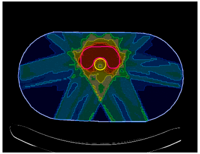
with M. Bangert @ DKFZ Heidelberg

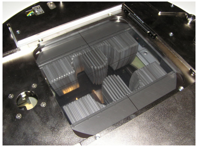
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

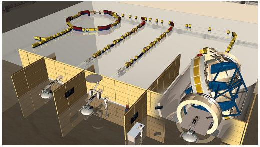
Max Planck Institute for Intelligent Systems

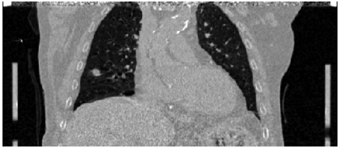
images: wikipedia / DKFZ











radiation treatment planning involves **approximately optimizing** an **imprecise** function subject to **uncertainties**.

DEUTSCHES KREBSFORSCHUNGSZENTRUM IN DER HELMHOLTZ-GEMEINSCHAFT

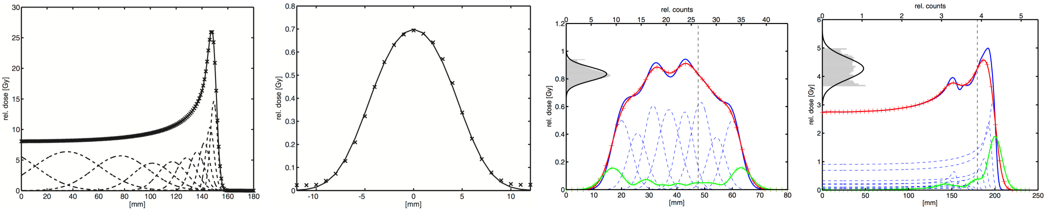
Propagating Uncertainty through Pipelines

Analytical Probabilistic Treatment Planning — with DKFZ Heidelberg

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

[Bangert et al., PMB, 2013, 2016, 2017]



- map all involved non-linear functions into tractable (Hilbert-) space, with **quality guarantees**, bounds on approximation error

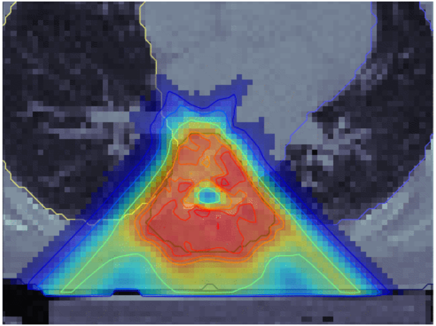
Propagating Uncertainty through Pipelines

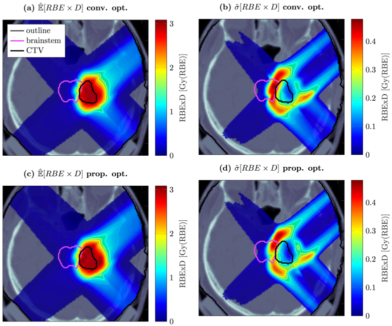
Analytical Probabilistic Treatment Planning — with DKFZ Heidelberg

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

[Bangert et al., PMB, 2013, 2016, 2017]





- map all involved non-linear functions into tractable (Hilbert-) space, with **quality guarantees**, bounds on approximation error
- track and **optimize uncertainties** across computation
- to improve treatment outcome, reduce risk of complications

Estimate z from computations c, under model m.

Prior: structural knowledge reduces complexity

Likelihood: modelling imprecision stabilizes algorithms

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

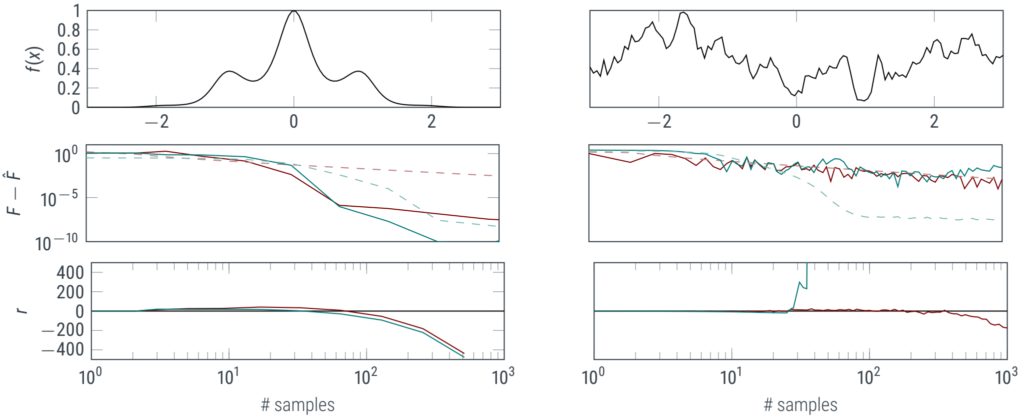
Posterior: tracking uncertainty for robustness

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]



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

Summary

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

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

Building numerical methods for contemporary challenges amounts to designing probabilistic models.

prior: structural knowledge reduces complexity

likelihood: imprecise computation lowers cost

posterior: uncertainty can be propagated through computations

evidence: model mismatch is detectable at run-time

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

Probabilistic Numerics — Uncertainty in Computation
Hennig, Osborne, Girolami Cambridge University Press, ETA 2019