Probabilistic Numerics

Uncertainty in Computation

Philipp Hennig

ParisBD
9 May 2017

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

MAX-PLANCK-GESELLSCHAFT

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

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, BFGS, Frank-Wolfe, …
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 their most 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 needs to build its own 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, ...]

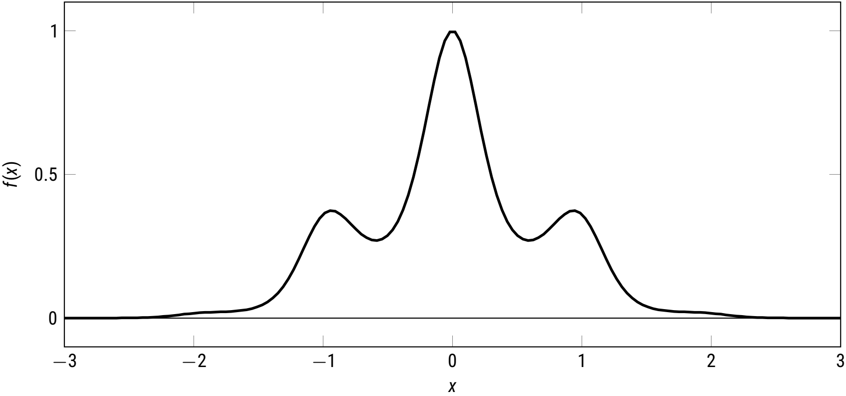
Numerical methods **estimate latent** quantities **given** the result of computations.

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



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

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

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

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

…conditioned on actively collected information …

computation as the collection of information

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

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

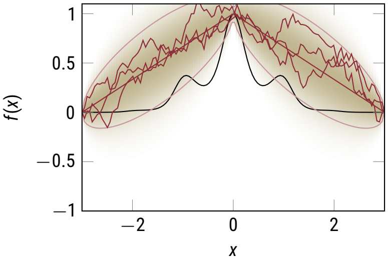
computation as the collection of information

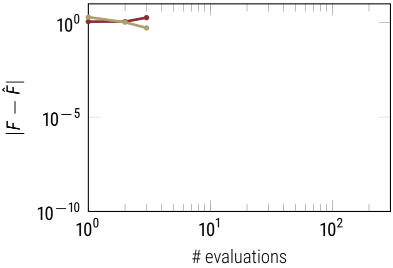
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

computation as the collection of information



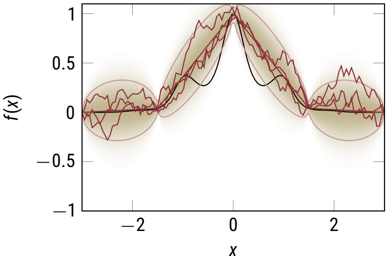


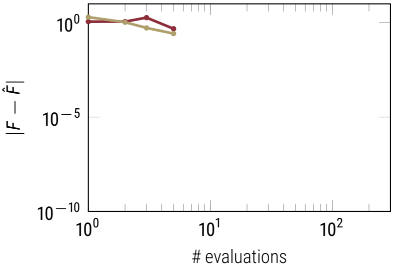
$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1,\dots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

…conditioned on actively collected information …

computation as the collection of information



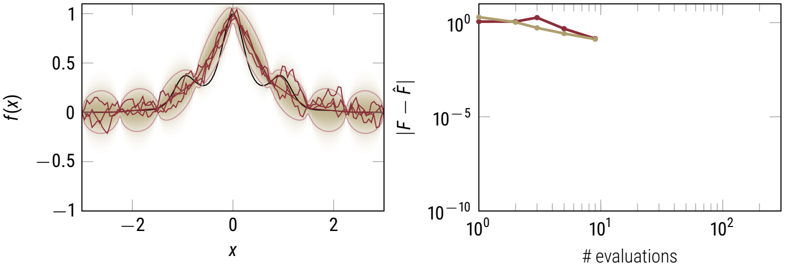


$$x_t = \arg\min \left[\mathrm{var}_{p(F|x_1,\dots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

...conditioned on actively collected information ...

computation as the collection of information

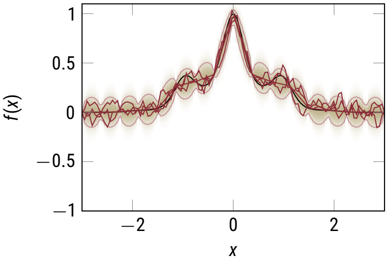


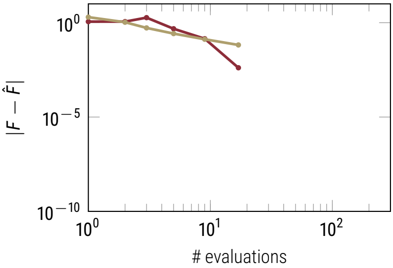
$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

computation as the collection of information



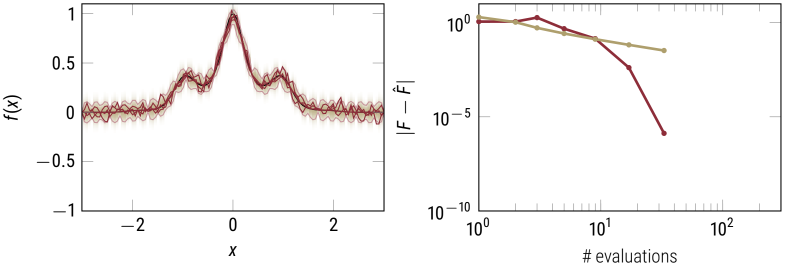


$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

…conditioned on actively collected information …

computation as the collection of information

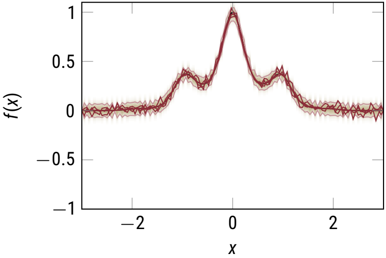


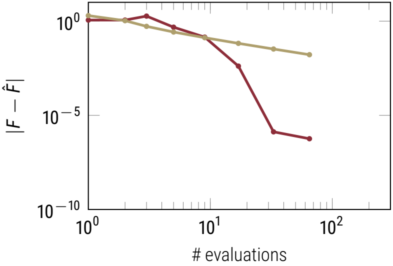
$$x_t = \arg\min \left[\mathrm{var}_{p(F|x_1,\dots,x_{t-1})}(F)\right]$$

- maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

computation as the collection of information



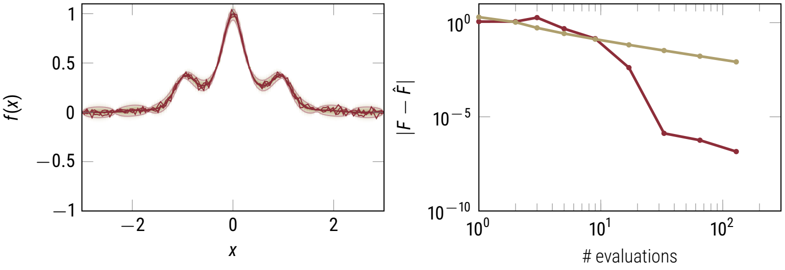


$$x_t = \arg\min \left[\mathrm{var}_{p(F \mid x_1, \dots, x_{t-1})}(F) \right]$$

- ✦ maximal reduction of variance yields regular grid

. . . conditioned on actively collected information . . .

computation as the collection of information

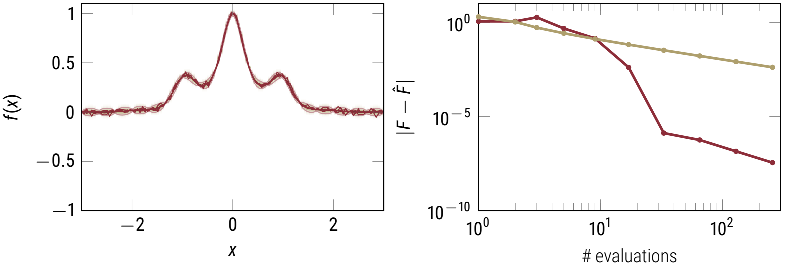


$$x_t = \arg\min \left[\operatorname{var}_{p(F \mid x_1, \ldots, x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

…conditioned on actively collected information …

computation as the collection of information

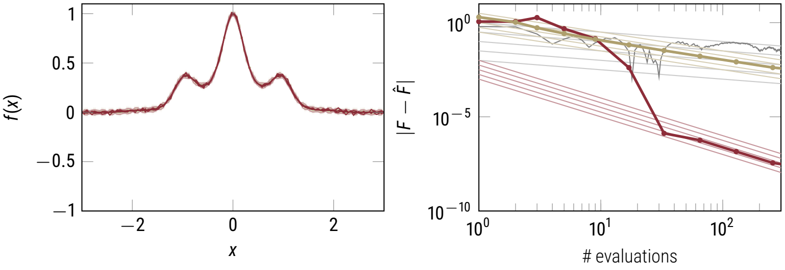


$$x_t = \arg\min \left[\operatorname{var}_{p(F|x_1,\ldots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

...conditioned on actively collected information ...

computation as the collection of information

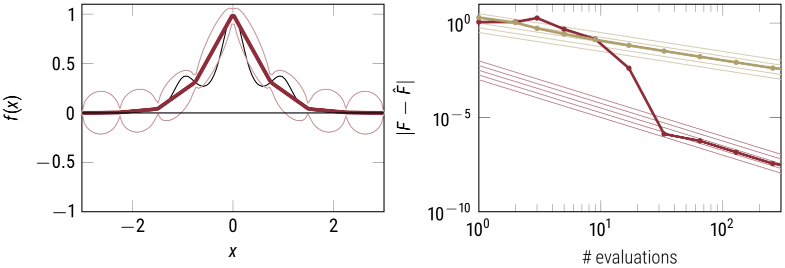


$$x_t = \arg\min \left[\operatorname{var}_{p(F|x_1,\dots,x_{t-1})}(F) \right]$$

- maximal reduction of variance yields regular grid

...yields the trapezoid rule!

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



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

- **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

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 ⟷ GP Regression

Linear Algebra [Hennig 2014]

Conjugate Gradients ⟷ Gaussian Regression

Nonlinear Optimization [Hennig & Kiefel 2013]

BFGS / Quasi-Newton ⟷ Autoregressive Filtering

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

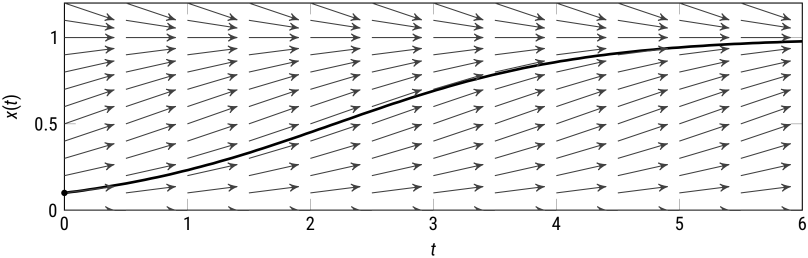
Differential Equations

Runge-Kutta; Nordsieck Methods ⟷ Gauss-Markov Filters

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** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

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** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

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** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

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** $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$

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 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?

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

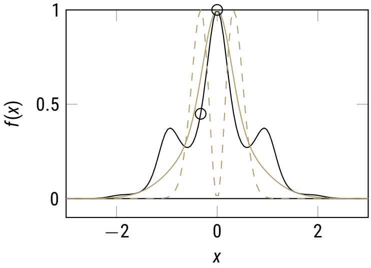
- $f > 0$ (f is probability measure)
- $f \propto \exp(-x^2)$ (f is product of prior and likelihood terms)
- $f \in \mathcal{C}^\infty$ (f is smooth)

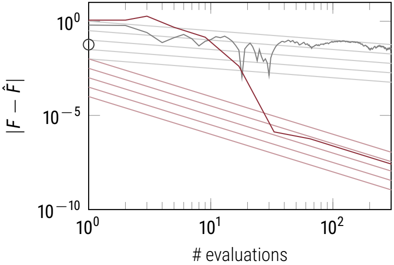
Explicit prior knowledge yields reduces complexity.

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

An integration prior for probability measures

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





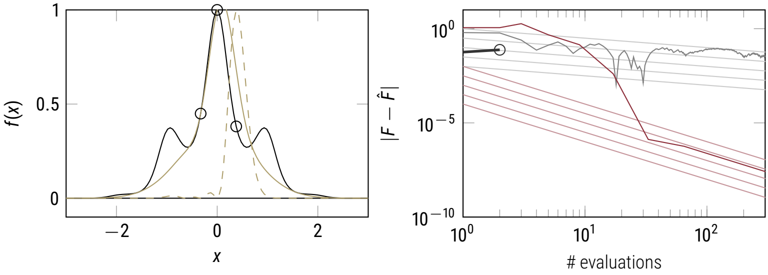
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

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]



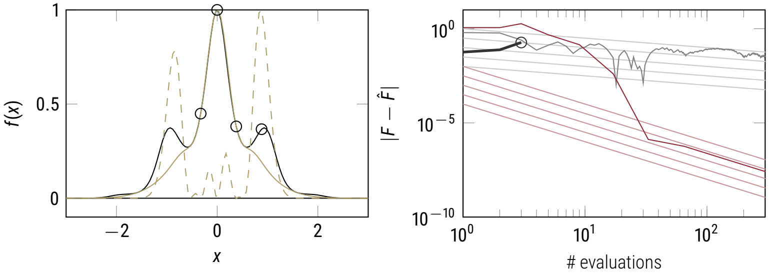
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

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]



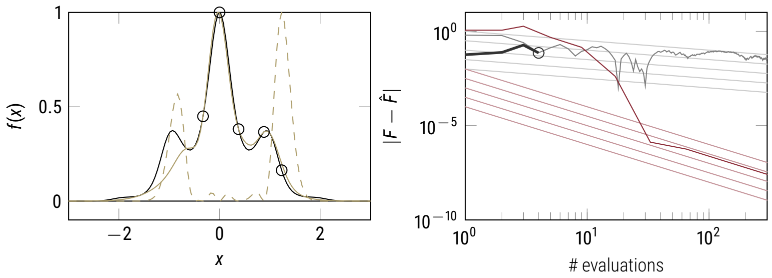
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

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]



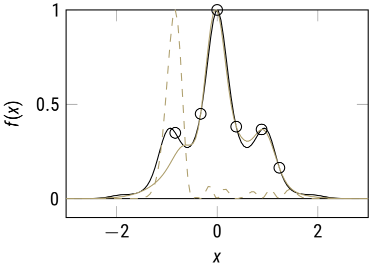
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

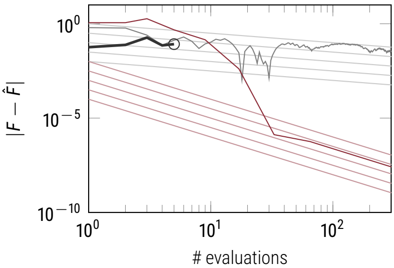
Explicit prior knowledge yields reduces complexity.

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





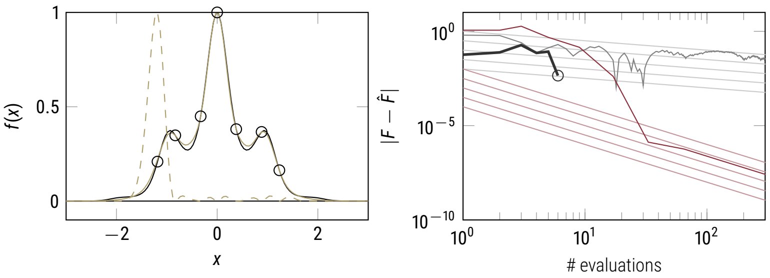
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

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]



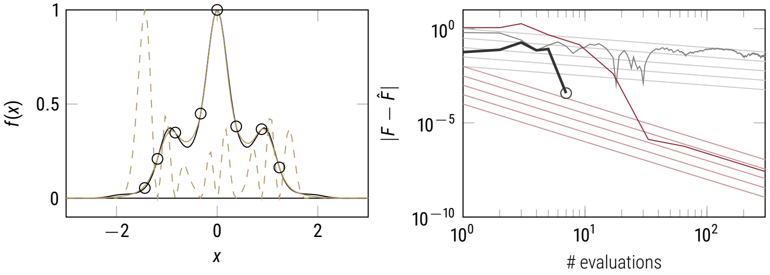
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

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]



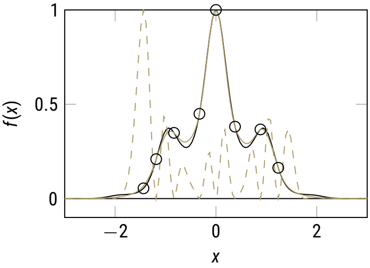
- adaptive node placement
- scales to, in principle, arbitrary dimensions
- faster (in wall-clock time) than MCMC

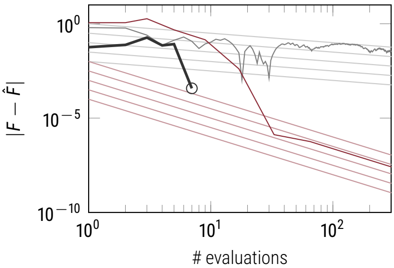
Explicit prior knowledge yields reduces complexity.

[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]

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:

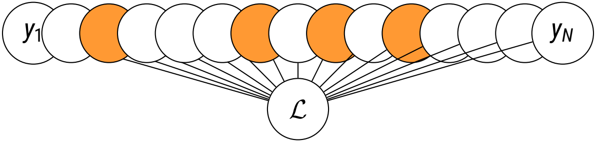
New numerics for Big Data

Uncertainty on Inputs directly effecting numerical decisions

In Big Data setting, batching introduces (Gaussian) noise

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

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



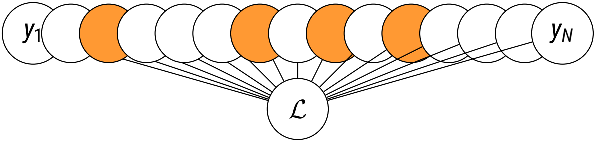
New numerics for Big Data

Uncertainty on Inputs directly effecting numerical decisions

In Big Data setting, batching introduces (Gaussian) noise

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

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



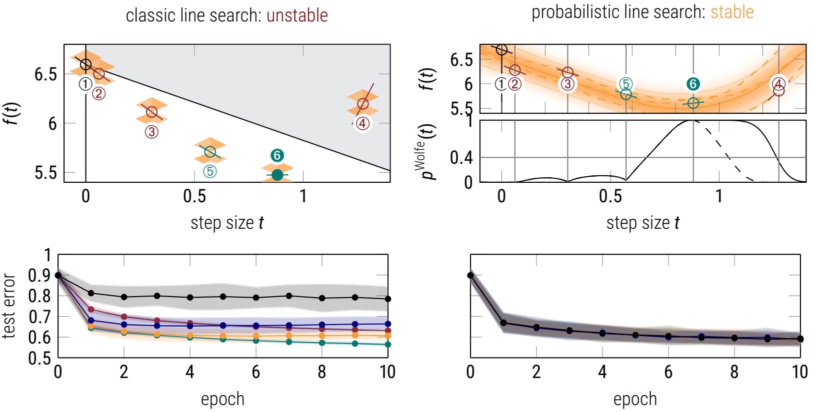
Classic methods are unstable to noise. E.g.: step size selection

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

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** [Balles & Hennig, arXiv 1612.05086]
- **early stopping** [Mahsereci, Balles & Hennig, arXiv 1703.09580]

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

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

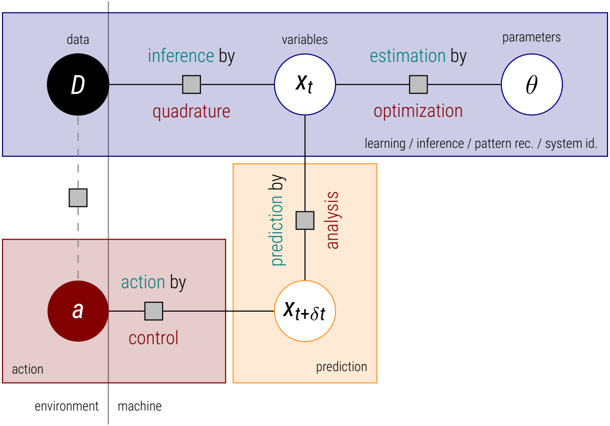
Posterior: tracking uncertainty for robustness

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

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

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

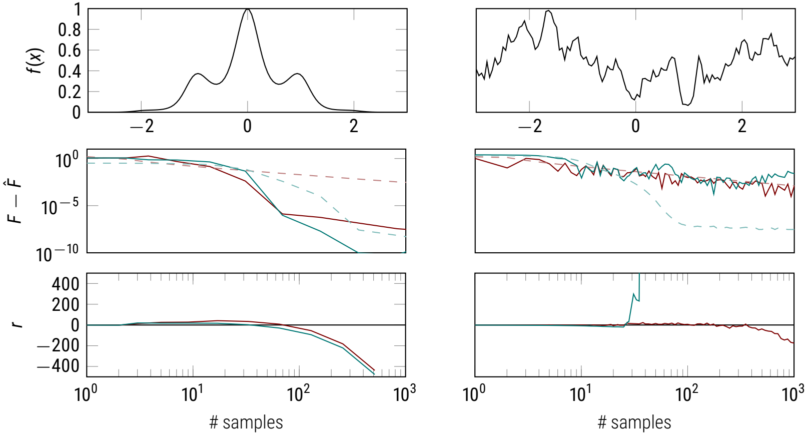
Posterior: tracking uncertainty for robustness

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]



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

Summary

Uncertain computation **as** and **for** 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

Specialized numerical methods **for** the challenges of machine learning can be developed within the conceptual framework **of** machine learning.

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

dummy slide

for use of pdf elsewhere