



UPPSALA
UNIVERSITET

Constructing stochastic quasi-Newton algorithms using Gaussian processes

Thomas Schön
Uppsala University
Sweden.

Joint work with **Adrian Wills**, University of Newcastle, Australia.

Svenska OA-konferensen (SOAK), October 23, 2019.

Mindset – Numerical methods are inference algorithms

A numerical method **estimates** a certain **latent** property **given** the result of computations.

Basic numerical methods and basic statistical models are **deeply connected in formal ways!**

Poincaré, H. **Calcul des probabilités**. Paris: Gauthier-Villars, 1896.

Diaconis, P. **Bayesian numerical analysis**. *Statistical decision theory and related topics*, IV(1), 163–175, 1988.

O'Hagan, A. **Some Bayesian numerical analysis**. *Bayesian Statistics*, 4, 345–363, 1992.

Hennig, P., Osborne, M. A., and Girolami, M. **Probabilistic numerics and uncertainty in computations**. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179), 2015.

Mindset – Numerical methods are inference algorithms

The task of a numerical algorithm is

to estimate unknown quantities from known ones.

Ex) basic algorithms that are equivalent to Gaussian MAP inference:

- Conjugate Gradients for linear algebra
- BFGS for nonlinear optimization
- Gaussian quadrature rules for integration
- Runge-Kutta solvers for ODEs

The structure of num. algs. is similar to statistical inference where

- The **tractable quantities** play the role of **"data" / "observations"**.
- The **intractable quantities** relate to **"latent" / "hidden"** quantities.

Problem formulation

We can use this relationship to derive new algorithms.

What? Solve the non-convex stochastic optimization problem

$$\min_{\theta} f(\theta)$$

when we only have access to **noisy** evaluations of $f(\theta)$ and its derivatives.

Why? These stochastic optimization problems are common:

- When the cost function cannot be evaluated on the entire dataset.
- When numerical methods approximate $f(\theta)$ and $\nabla^i f(\theta)$.
- ...

How? – our contribution for today

How? Learn a probabilistic nonlinear model of the Hessian.

Provides a local approximation of the cost function $f(\theta)$.

Use this local model to compute a search direction.

Stochastic line search. We have derived a stochastic version of the Wolfe/Armijo condition allowing for stochastic line search.

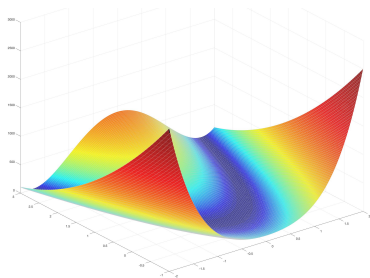
Captures second-order information (curvature) which opens up for better performance compared to a pure gradient-based method.

Intuitive preview example – Rosenbrock's banana function

$$\text{Let } f(\theta) = (1 - \theta_1)^2 + 100(\theta_2 - \theta_1^2)^2.$$

Deterministic problem

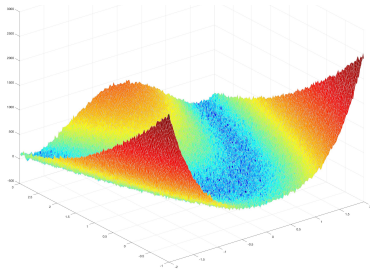
$$\min_{\theta} f(\theta)$$



Stochastic problem

$$\min_{\theta} f(\theta)$$

when we only have access to noisy versions of the cost function ($\tilde{f}(\theta) = f(\theta) + e$, $e = \mathcal{N}(0, 30^2)$) and its noisy gradients.



Aim: Derive a stochastic quasi-Newton algorithm.

Spin-off: Combine it with particle filters for maximum likelihood identification in nonlinear dynamical models.

1. Mindset (probabilistic numerics) and problem formulation
2. **A non-standard take on quasi-Newton**
3. μ on the Gaussian Process (GP)
4. Assembling a new stochastic optimization algorithm
 - a. Representing the Hessian with a Gaussian process
 - b. Learning the Hessian
5. Testing ground – maximum likelihood for nonlinear dynamics
6. New AI initiative at Uppsala University

Quasi-Newton – A non-standard take

Our problem is of the form

$$\min_{\theta} f(\theta)$$

Idea underlying (quasi-)Newton methods: Learn a local quadratic model $q(\theta_k, \delta)$ of the cost function $f(\theta)$ around the current iterate θ_k

$$q(\theta_k, \delta) = f(\theta_k) + g(\theta_k)^\top \delta + \frac{1}{2} \delta^\top H(\theta_k) \delta$$

$$g(\theta_k) = \nabla f(\theta) \Big|_{\theta=\theta_k}, \quad H(\theta_k) = \nabla^2 f(\theta) \Big|_{\theta=\theta_k}, \quad \delta = \theta - \theta_k.$$

We have measurements of

- the cost function $f_k = f(\theta_k)$,
- and its gradient $g_k = g(\theta_k)$.

Question: How do we update the Hessian model?

Useful basic facts

Line segment connecting two adjacent iterates θ_k and θ_{k+1} :

$$r_k(\tau) = \theta_k + \tau(\theta_{k+1} - \theta_k), \quad \tau \in [0, 1].$$

1. The **fundamental theorem of calculus** states that

$$\int_0^1 \frac{\partial}{\partial \tau} \nabla f(r_k(\tau)) d\tau = \nabla f(r_k(1)) - \nabla f(r_k(0)) = \underbrace{\nabla f(\theta_{k+1})}_{g_{k+1}} - \underbrace{\nabla f(\theta_k)}_{g_k}.$$

2. The **chain rule** tells us that

$$\frac{\partial}{\partial \tau} \nabla f(r_k(\tau)) = \nabla^2 f(r_k(\tau)) \frac{\partial r_k(\tau)}{\partial \tau} = \nabla^2 f(r_k(\tau)) (\theta_{k+1} - \theta_k).$$

$$\underbrace{g_{k+1} - g_k}_{=y_k} = \int_0^1 \frac{\partial}{\partial \tau} \nabla f(r_k(\tau)) d\tau = \int_0^1 \nabla^2 f(r_k(\tau)) d\tau \underbrace{(\theta_{k+1} - \theta_k)}_{s_k}.$$

Result – the quasi-Newton integral

With the definitions $y_k \triangleq g_{k+1} - g_k$ and $s_k \triangleq \theta_{k+1} - \theta_k$ we have

$$y_k = \int_0^1 \nabla^2 f(r_k(\tau)) d\tau s_k.$$

Interpretation: The difference between two consecutive gradients (y_k) constitute a **line integral observation of the Hessian**.

Problem: Since the Hessian is unknown there is no functional form available for it.

Solution 1 – recovering existing quasi-Newton algorithms

Existing quasi-Newton algorithms (e.g. BFGS, DFP, Broyden's method) assume the Hessian to be constant

$$\nabla^2 f(r_k(\tau)) \approx H_{k+1}, \quad \tau \in [0, 1],$$

implying the following approximation of the integral (**secant condition**)

$$y_k = H_{k+1} s_k.$$

Find H_{k+1} by **regularizing** H :

$$\begin{aligned} H_{k+1} &= \min_H \|H - H_k\|_W^2, \\ \text{s.t. } & H = H^\top, \quad H s_k = y_k, \end{aligned}$$

Equivalently, the existing quasi-Newton methods can be interpreted as **particular instances of Bayesian linear regression**.

Solution 2 – use a flexible nonlinear model

The approach used here is fundamentally different.

Recall that the problem is **stochastic** and **nonlinear**.

Hence, we need a model that can deal with such a problem.

Idea: Represent the Hessian using a **Gaussian process** learnt from data.

μ on the Gaussian process (GP)

The Gaussian process is a model for nonlinear functions

Q: Why is the Gaussian process used everywhere?

It is a **non-parametric** and **probabilistic** model for nonlinear functions.

- **Non-parametric** means that it does not rely on any particular parametric functional form to be postulated.
- **Probabilistic** means that it takes uncertainty into account in every aspect of the model.

An abstract idea

In probabilistic (Bayesian) linear regression

$$y_t = \underbrace{\theta^T \mathbf{x}_t}_{f(\mathbf{x}_t)} + e_t, \quad e_t \sim \mathcal{N}(0, \sigma^2),$$

we place a prior on θ , e.g. $\theta \sim \mathcal{N}(0, \alpha^2 I)$.

(Abstract) idea: What if we instead place a prior directly on the function $f(\cdot)$

$$f \sim p(f)$$

and look for $p(f | y_{1:T})$ rather than $p(\theta | y_{1:T})$?!

One concrete construction

Well, one (arguably simple) idea on how we can reason probabilistically about an unknown function f is by assuming that $f(\mathbf{x})$ and $f(\mathbf{x}')$ are jointly Gaussian distributed

$$\begin{pmatrix} f(\mathbf{x}) \\ f(\mathbf{x}') \end{pmatrix} \sim \mathcal{N}(m, K).$$

If we accept the above idea we can without conceptual problems generalize to any *arbitrary* finite set of input values $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$.

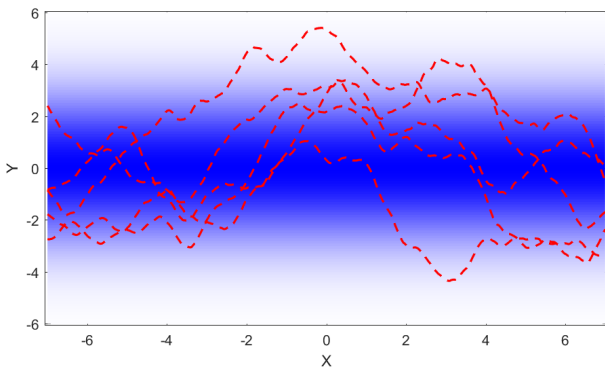
$$\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_T) \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_T) \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_T) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_T, \mathbf{x}_1) & \dots & k(\mathbf{x}_T, \mathbf{x}_T) \end{pmatrix} \right)$$

Definition: (Gaussian Process, GP) A GP is a (potentially infinite) collection of random variables such that any finite subset of it is jointly distributed according to a Gaussian.

We now have a prior!

$$f \sim \mathcal{GP}(m, k)$$

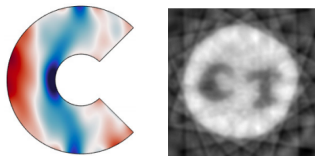
The GP is a **generative** model so let us first sample from the prior.



Snapshot — Constrained GP for tomographic reconstruction

Tomographic reconstruction goal: Build a map of an unknown quantity within an object using information from irradiation experiments.

- Ex1) Modelling and reconstruction of strain fields.
- Ex2) Reconstructing the internal structure from limited x-ray projections.



Carl Jidling, Johannes Hendriks, Niklas Wahlström, Alexander Gregg, TS, Chris Wensrich and Adrian Wills. **Probabilistic modelling and reconstruction of strain.** *Nuclear inst. and methods in physics research: section B*, 436:141-155, 2018.

Zenith Purisha, Carl Jidling, Niklas Wahlström, Simo Särkkä and TS. **Probabilistic approach to limited-data computed tomography reconstruction.** *Inverse Problems*, 35(10):105004, 2019.

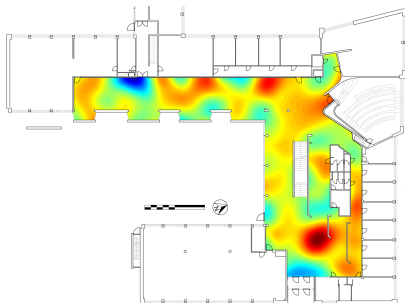
Carl Jidling, Niklas Wahlström, Adrian Wills and TS. **Linearly constrained Gaussian processes.** *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, December, 2017.

Snapshot — Model of the ambient magnetic field with GPs

The Earth's magnetic field sets a background for the ambient magnetic field. Deviations make the field vary from point to point.

Aim: Build a map (i.e., a model) of the magnetic environment based on magnetometer measurements.

Solution: Customized Gaussian process that obeys Maxwell's equations.



www.youtube.com/watch?v=enlMiUqPVJo

Arno Solin, Manon Kok, Niklas Wahlström, TS and Simo Särkkä. **Modeling and interpolation of the ambient magnetic field by Gaussian processes.** *IEEE Transactions on Robotics*, 34(4):1112–1127, 2018.

Stochastic optimization

Stochastic quasi-Newton integral

$$y_k = \int_0^1 \underbrace{B(r_k(\tau))}_{=\nabla^2 f(r_k(\tau))} s_k d\tau + e_k,$$

corresponds to noisy (e_k) gradient observations.

Since $B(\mathbf{x})s_k$ is a column vector, the integrand is given by

$$\text{vec}(B(\mathbf{x})s_k) = (s_k^T \otimes I) \text{vec}(B(\mathbf{x})) = (s_k^T \otimes I) \text{vec}(B(\mathbf{x})),$$

where $\text{vec}(B(\mathbf{x})) = \underbrace{D \text{vech}(B(\mathbf{x}))}_{\tilde{B}(\mathbf{x})}$.

Let us use a GP model for the unique elements of the Hessian

$$\tilde{B}(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}')).$$

Resulting stochastic qN integral and Hessian model

Summary: resulting stochastic quasi-Newton integral:

$$y_k = D_k \int_0^1 \tilde{B}(r_k(\tau)) d\tau + e_k,$$

with the following model for the Hessian

$$\tilde{B}(\theta) \sim \mathcal{GP}(\mu(\theta), \kappa(\theta, \theta')).$$

The Hessian can now be estimated using tailored GP regression.

Linear operators (such as a line integral or a derivative) acting on a GP results in a another GP.

Resulting stochastic optimization algorithm

Standard numerical optimization loop with **non-standard components**.

Algorithm 1 Stochastic optimization

1. **Initialization** ($k = 1$)
 2. **while** *not terminated* **do**
 - (a) Compute a **search direction** p_k using the current approximation of the gradient g_k and Hessian B_k .
 - (b) **Stochastic line search** to find a step length α_k and set
$$\theta_{k+1} = \theta_k + \alpha_k p_k.$$
 - (c) **Update the Hessian model** (tailored GP regression).
 - (d) Set $k := k + 1$.
 3. **end while**
-

Derived a stochastic Armijo condition enabling stochastic line search.

Testing ground – nonlinear sys.id.

Probabilistic modelling of dynamical systems

$$x_t = f(x_{t-1}, \theta) + w_t,$$

$$y_t = g(x_t, \theta) + e_t,$$

$$x_0 \sim p(x_0 | \theta),$$

$$(\theta \sim p(\theta)).$$

$$x_t | (x_{t-1}, \theta) \sim p(x_t | x_{t-1}, \theta),$$

$$y_t | (x_t, \theta) \sim p(y_t | x_t, \theta),$$

$$x_0 \sim p(x_0 | \theta),$$

$$(\theta \sim p(\theta)).$$

Corresponding full probabilistic model:

$$p(x_{0:T}, \theta, y_{1:T}) = \prod_{t=1}^T \underbrace{p(y_t | x_t, \theta)}_{\text{observation}} \underbrace{\prod_{t=1}^T p(x_t | x_{t-1}, \theta)}_{\text{dynamics}} \underbrace{p(x_0 | \theta)}_{\text{state}} \underbrace{p(\theta)}_{\text{param.}}$$

prior

Model = probability distribution!

Maximum likelihood – model the unknown parameters as a deterministic variable θ and solve

$$\max_{\theta} p(y_{1:T} | \theta),$$

Challenge: The optimization problem is stochastic!

Cost function – the likelihood

Each element $p(y_t | y_{1:t-1}, \theta)$ in the likelihood

$$p(y_{1:T} | \theta) = \prod_{t=1}^T p(y_t | y_{1:t-1}, \theta),$$

can be computed by averaging over all possible values for the state x_t ,

$$p(y_t | y_{1:t-1}, \theta) = \int p(y_t | x_t, \theta) \underbrace{p(x_t | y_{1:t-1}, \theta)}_{\text{approx. by PF}} dx_t.$$

Non-trivial fact: The likelihood estimates obtained from the particle filter (PF) are **unbiased**.

Tutorial paper on the use of the PF (an instance of sequential Monte Carlo, SMC) for nonlinear system identification

ex) Simple linear toy problem

Identify the parameters $\theta = (a, c, q, r)^\top$ in

$$x_{t+1} = ax_t + w_t,$$

$$y_t = cx_t + e_t,$$

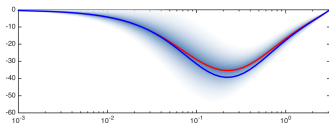
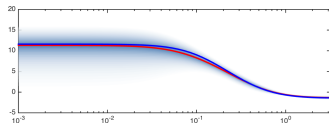
$$w_t \sim \mathcal{N}(0, q^2),$$

$$e_t \sim \mathcal{N}(0, r^2).$$

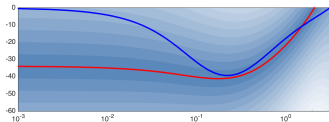
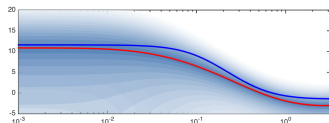
Observations:

- The likelihood $L(\theta) = p(y_{1:T} | \theta)$ and its gradient $\nabla_{\theta} L(\theta)$ are available in closed form via standard Kalman filter equations.
- Standard gradient-based search algorithms applies.
- Deterministic optimization problem $(L(\theta), \nabla_{\theta} L(\theta))$ noise-free).

ex) Simple linear toy problem



Both alg. in the noise-free case.



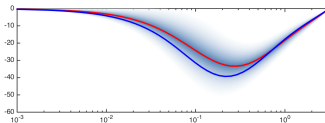
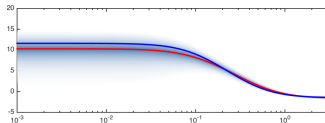
Classical BFGS alg. for noisy observations of $L(\theta)$ and $\nabla L(\theta)$.

100 independent datasets.

Clear blue – True system

Red – Mean value of estimate

Shaded blue – individual results



GP-based BFGS alg. with noisy observations of $L(\theta)$ and $\nabla L(\theta)$. 27/36

ex) Nonlinear system

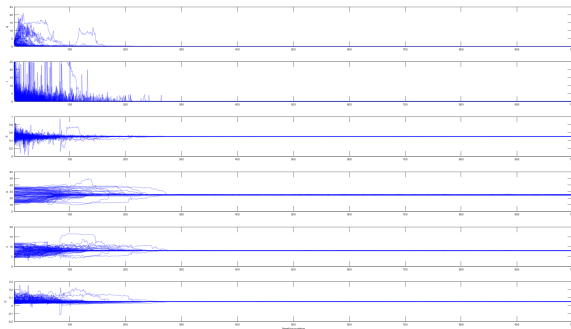
Identify the parameters $\theta = (a, c, d, q, r)^T$ in

$$x_{t+1} = ax_t + b \frac{x_t}{1+x_t^2} + c \cos(1.2t) + w_t,$$

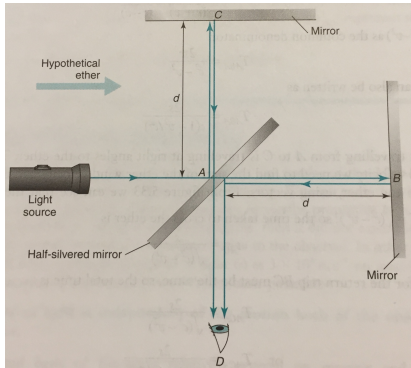
$$w_t \sim \mathcal{N}(0, q^2),$$

$$y_t = dx_t^2 + e_t,$$

$$e_t \sim \mathcal{N}(0, r^2).$$



ex) Laser interferometry



The classic Michelson-Morley experiment from 1887.

Idea: Merge two light sources to create an interference pattern by superposition.

Two cases:

1. Mirror B and C at the **same** distance from mirror A.
2. Mirror B and C at **different** distances from mirror A.

ex) Laser interferometry

Dynamics: constant velocity model (with unknown force w)

$$\begin{pmatrix} \dot{p} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} p \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ w \end{pmatrix}.$$

Measurements: generated using two detectors

$$y_1 = \alpha_0 + \alpha_1 \cos(\kappa p) + e_1, \quad e_1 \sim \mathcal{N}(0, \sigma^2),$$

$$y_2 = \beta_0 + \beta_1 \sin(\kappa p + \gamma) + e_2, \quad e_2 \sim \mathcal{N}(0, \sigma^2).$$

Unknown parameters: $\theta = (\alpha_0 \quad \alpha_1 \quad \beta_0 \quad \beta_1 \quad \gamma \quad \sigma)^T$.

Resulting maximum likelihood system identification problem

$$\max_{\theta} p(y_{1:T} | \theta)$$

ex) Laser interferometry

Scaling up to large(r) problems

What is the key limitation of our GP-based optimization algorithm?

It **does not** scale to large-scale problems!

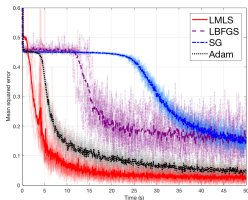
Still highly useful and competitive for **small to medium** sized problems.

We have developed a **new** technique that scales to **large(r)** problems.

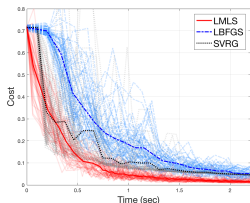
Scaling up to large(r) problems

Key innovations:

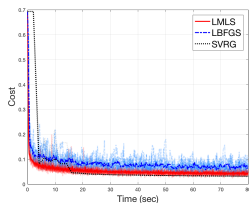
- Replace the GP with a matrix updated using fast Cholesky routines.
- Exploit a receding history of iterates and gradients akin to L-BFGS.
- Same stochastic line search applicable.



Training a deep CNN for MNIST data.



Logistic loss function with an L2 regularizer, gisette, 6 000 observations and 5 000 unknown variables.



Logistic loss function with an L2 regularizer, URL, 2 396 130 observations and 3 231 961 unknown variables.

The **heart** of this plan consists in a **time-limited five year effort** consisting of an **antidisciplinary entity** doing **AI for the sciences.**

We are building a **collocated team**:

- Work together on a regular basis during a longer time span.
- Create a reality where **trust** can be built.
- **AI internal sabbatical periods** lasting around 12 months.
- Hope that this facilitates a **rapid organic renewal** of the university.

We are growing!

Tenure track Assistant Professor

Large-scale (stochastic) optimization

Research intensive, **competitive package**, including 1 PhD student, 4 post-doc years and cash.

More information on my website soon!

Conclusions

Message: The Gaussian process can be used to construct **new algorithms** for stochastic optimization.

Derived the stochastic quasi-Newton integral.

Built a second-order model to approximate the cost function.

New **stochastic line search** is an important part of the solution.

Standard numerical optimization loop with **non-standard components**.

Testing ground — Probabilistic modelling of nonlinear state space models