



UPPSALA
UNIVERSITET

Assembling stochastic quasi-Newton algorithms using GPs

Thomas Schön, Uppsala University, Sweden.

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

27th ERNSI Workshop in System Identification, Pembroke College,
Cambridge, UK, September 25, 2018.

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.

probabilistic-numerics.org/

The Gaussian process (GP) is a natural tool to understand this and to open up for new algorithms to be constructed.

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? 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.

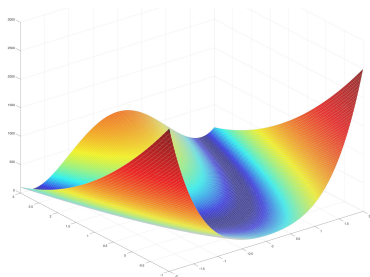
Captures second-order information (curvature) which opens up for better performance compared to a pure gradient-based method.

Intuitive preview example – Rosenbrock's banana function

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

Deterministic problem

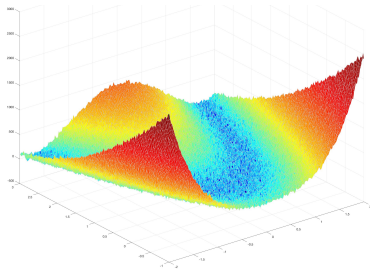
$$\min_{\theta} f(\theta)$$



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 state space models.

1. Mindset (probabilistic numerics) and problem formulation
2. **A non-standard take on quasi-Newton**
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 SSMs

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 noisy 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.

Two of the remaining challenges:

1. Can we use line integral observations when learning a GP?
2. How do we ensure that the resulting GP represents a Hessian?

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 unique elements of the Hessian

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

The Hessian can now be estimated using tailored GP regression.

Linear transformations (such as an integral or a derivative) of a GP results in a new 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**

Curvature information is useful also for stochastic optimization.

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

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

$$y_t = cx_t + e_t,$$

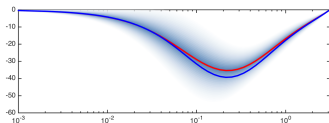
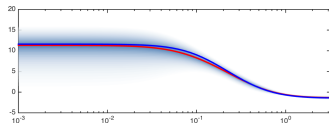
$$w_t \sim \mathcal{N}(0, q^2),$$

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

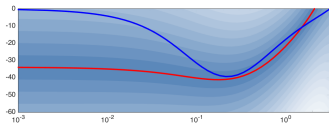
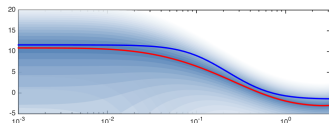
Observations:

- 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.



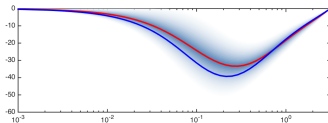
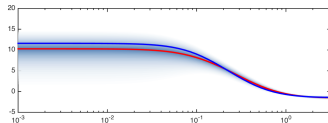
Classical BFGS alg. for noisy observations of $L(\theta)$ and $\nabla L(\theta)$.

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 $\nabla L(\theta)$. 17/21

ex) Nonlinear system

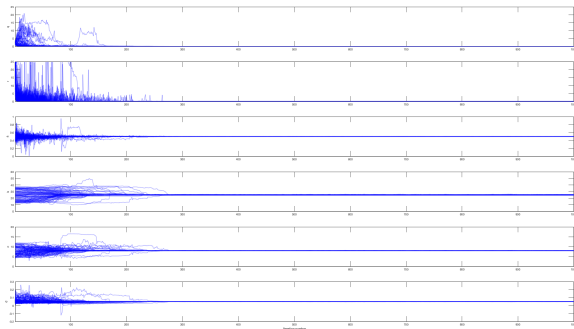
Identify the parameters $\theta = (a, c, d, q, r)^T$ in

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

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

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

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



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!

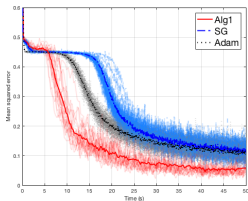
Still highly useful and competitive for **small to medium** sized problems.

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

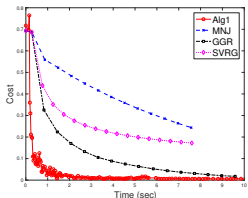
Scaling up to large(r) problems

Key innovations:

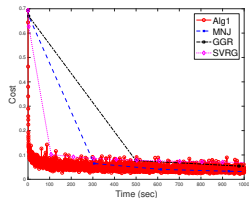
- Replace the GP with a matrix updated using fast Cholesky routines.
- Exploit a receding history of iterates and gradients akin to L-BFGS.
- New stochastic line search algorithm.



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.

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.

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

Testing ground — Probabilistic modelling of nonlinear state space models

We also have another technique that scales to large(r) problems.

Looking for new PhD students and Professors

Dave Zachariah and I are looking for 2 PhD students:

1. Probabilistic programming deployment of deep learning
 2. Causality
-

Wallenberg AI, Autonomous Systems and Software Program (WASP) is Sweden's largest ever individual research program.

Information about the project: wasp-sweden.org/

Information about the positions: wasp-sweden.org/wasp-ai-positions/

