UPPSALA
UNIVERSITET

On the Construction of Probabilistic Newton-Type Algorithms

Thomas Schön, Uppsala University

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

SIAM Conference on Uncertainty Quantification, Garden Grove, CA, USA.
April 18, 2018.

Problem formulation

What? Solve the non-convex stochastic optimization problem

$$\max_x f(x)$$

when we only have access to **noisy** evaluations of $f(x)$ 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 $f(x)$ and $\nabla^i f(x)$.
- ...

How? — our contribution

How? Learn a probabilistic nonlinear model of the Hessian.

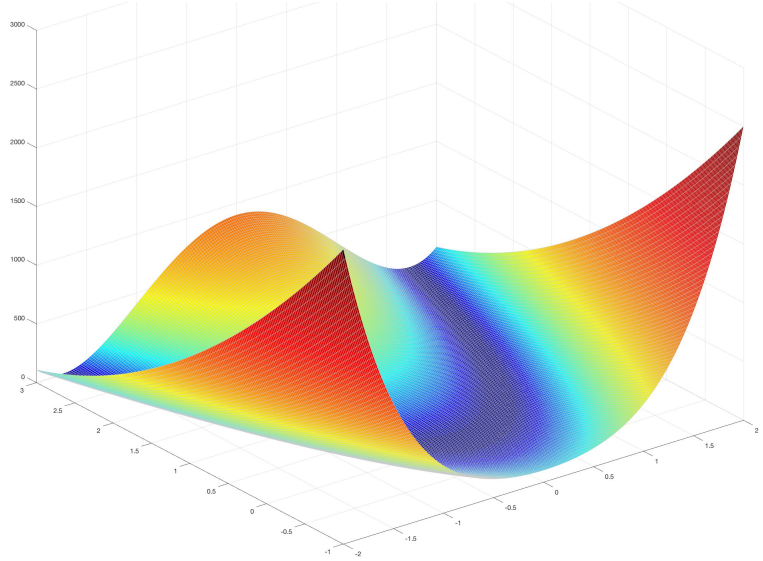
Provides a local approximation of the cost function $f(x)$.

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 function

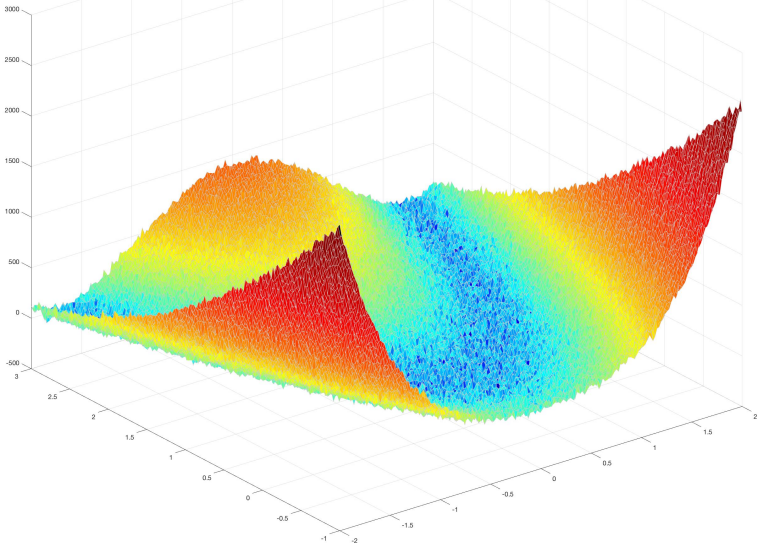
Let $f(x) = (a - x_1)^2 + b(x_2 - x_1^2)^2$, where $a = 1$ and $b = 100$.



Deterministic problem

$$\min_x f(x)$$

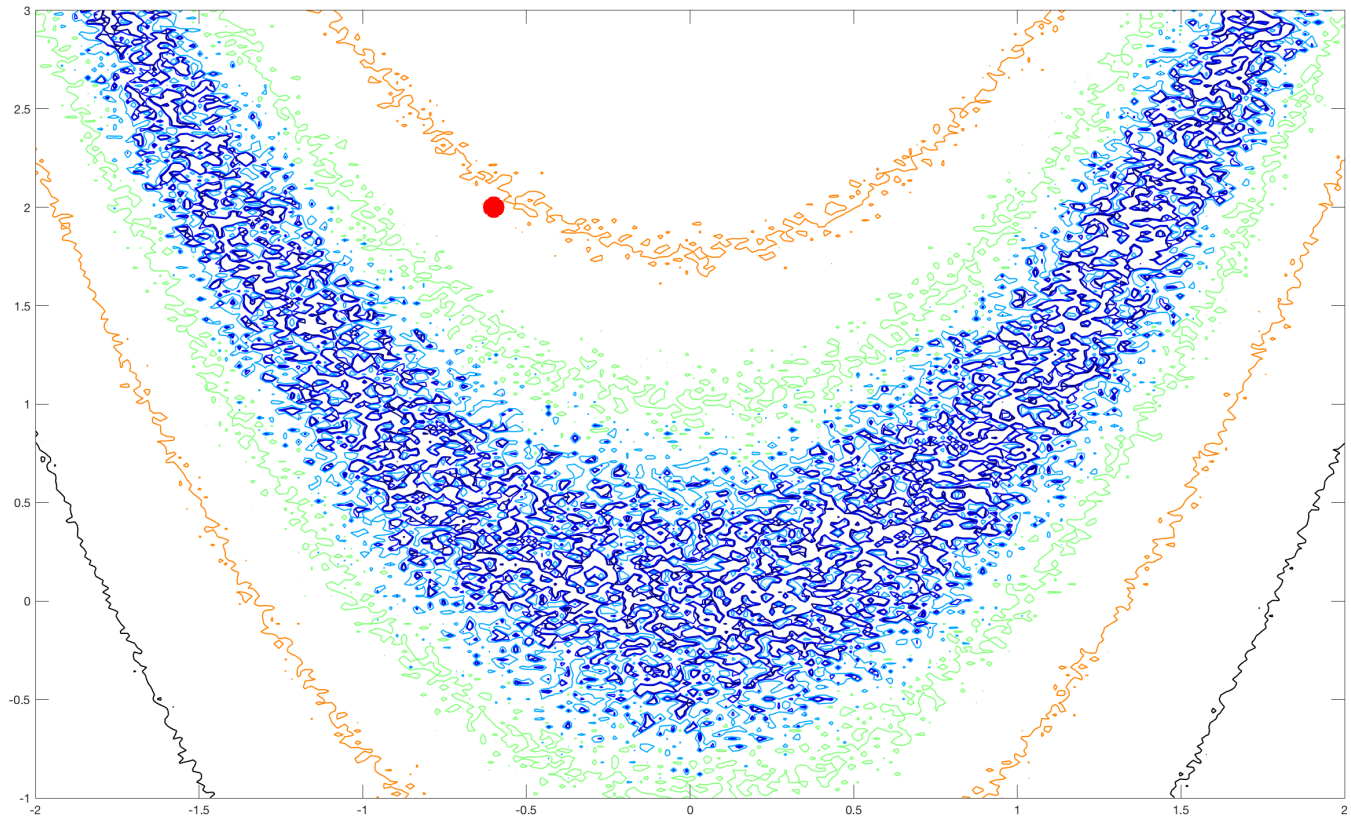
Stochastic problem



$$\min_x f(x)$$

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

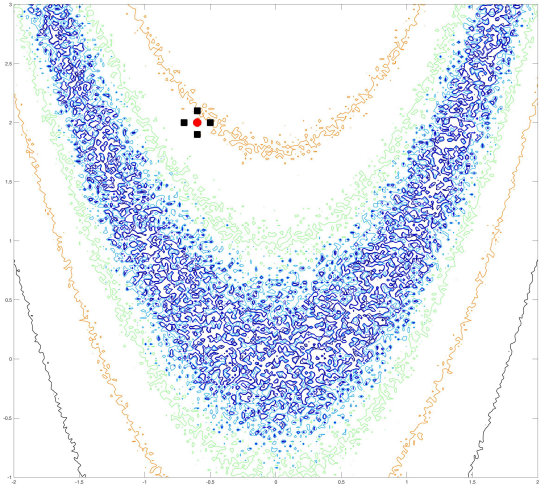
Adam at work



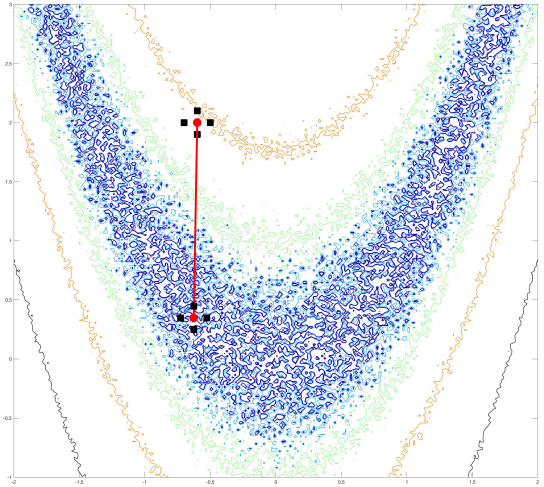
By not using the curvature information we expose ourself to the "banana-problem".

New algorithm at work — overall result

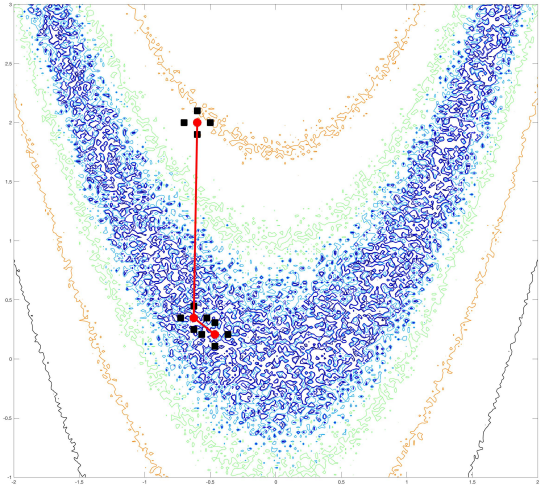
Initial value



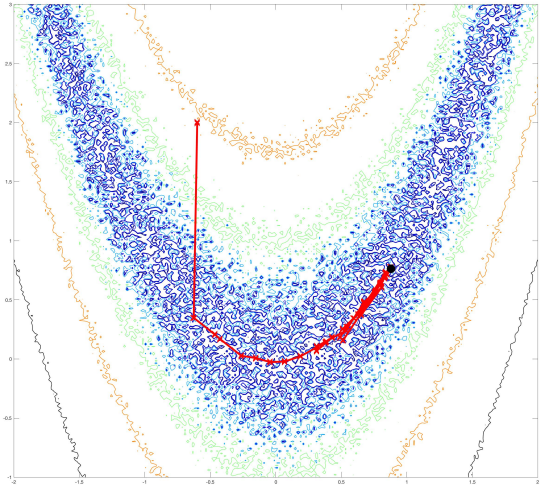
Iteration 1



Iteration 2



Iteration 50



Quasi-Newton — A non-standard take

Our problem is of the form

$$\max_x f(x)$$

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

$$q(x_k, \delta) = f(x_k) + g(x_k)^\mathsf{T}\delta + \frac{1}{2}\delta^\mathsf{T} H(x_k)\delta$$

A second-order Taylor expansion around x_k, where

$$g(x_k) = \nabla f(x)\big|_{x=x_k},$$
$$H(x_k) = \nabla^2 f(x)\big|_{x=x_k},$$
$$\delta = x - x_k.$$

Available data

We have measurements of the

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

Question: How do we update the Hessian model?

Line segment connecting two adjacent iterates x_k and x_{k+1}:

$$r_k(\tau) = x_k + \tau(x_{k+1} - x_k), \qquad \tau \in [0, 1].$$

Useful basic facts

The fundamental theorem of calculus states that

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

and the chain rule tells us that

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

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

Result — the quasi-Newton integral

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

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

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

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

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

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

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

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

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

Solution 2 — use a flexible nonlinear model

Our approach 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?

GP prior for the Hessian

Stochastic quasi-Newton integral

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

corresponds to noisy (e_k) gradient observations.

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

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

Resulting stochastic optimization algorithm

Standard non-convex 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
   $$x_{k+1} = x_k + \alpha_k p_k.$$
   (c) Set $k := k + 1$
   (d) Update the Hessian estimate (tailored GP regression)
3. **end while**

Maximum likelihood nonlinear system identification

$$
\begin{aligned}
x_t &= f(x_{t-1}, \theta) + w_t, \\
y_t &= g(x_t, \theta) + e_t, \\
x_0 &\sim p(x_0 \mid \theta), \\
(\theta &\sim p(\theta)).
\end{aligned}
\qquad
\begin{aligned}
x_t \mid (x_{t-1}, \theta) &\sim p(x_t \mid x_{t-1}, \theta), \\
y_t \mid (x_t, \theta) &\sim p(y_t \mid x_t, \theta), \\
x_0 &\sim p(x_0 \mid \theta), \\
(\theta &\sim p(\theta)).
\end{aligned}
$$

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

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

Challenge: The optimization problem is stochastic!

Cost function – the likelihood

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

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

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

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

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

Thomas B. Schön, Fredrik Lindsten, Johan Dahlin, Johan Wagberg, Christian A. Naesseth, Andreas Svensson and Liang Dai. **Sequential Monte Carlo methods for system identification**, *Proceedings of the 17th IFAC Symposium on System Identification (SYSID)*, Beijing, China, October 2015.

ex) Simple linear toy problem

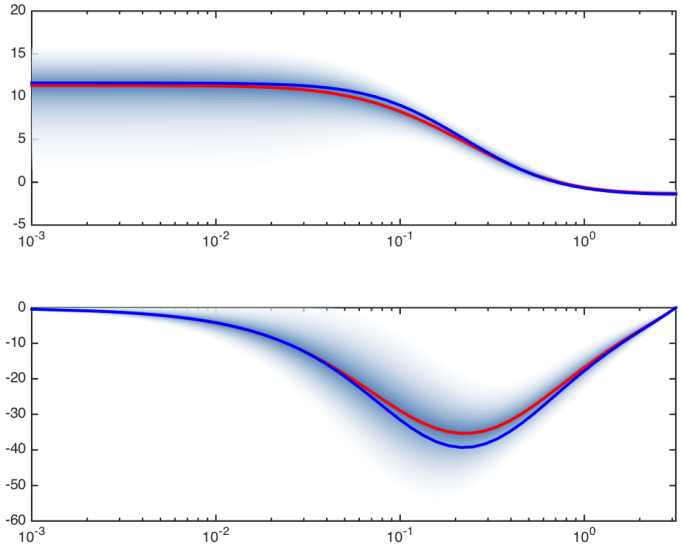
Identify the parameters $\theta = (a, c, q, r)^\mathsf{T}$ in

$$x_{t+1} = a x_t + w_t, \qquad w_t \sim \mathcal{N}(0, q),$$
$$y_t = c x_t + e_t, \qquad e_t \sim \mathcal{N}(0, r).$$

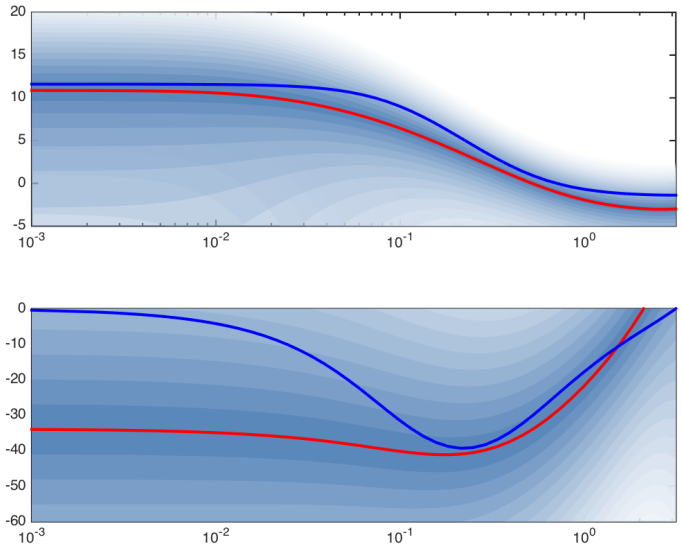
Observations:

- The likelihood $L(\theta) = p(y_{1:T} \mid \theta)$ 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.



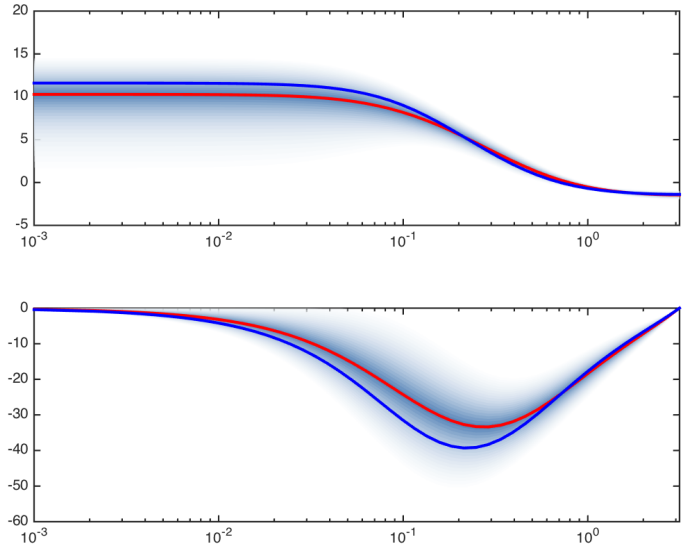
Classical BFGS alg. for noisy observations of $L(\theta)$ and $\nabla L(\theta)$.

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 $\nabla L(\theta)$.

Ongoing work – scaling up to large problems

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

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

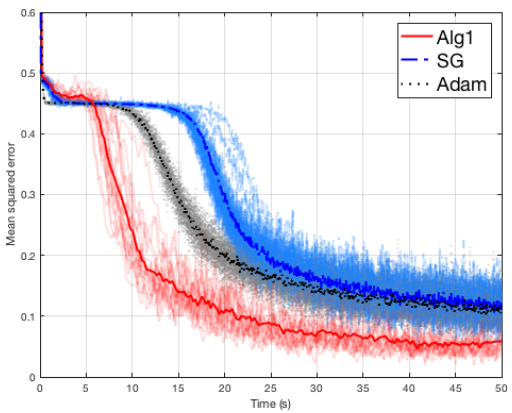
It is still highly useful and competitive for **small to medium** sized problems involving up to a coupled of hundred parameters or so.

We have developed a **new** technique that scales to **very large** problems.

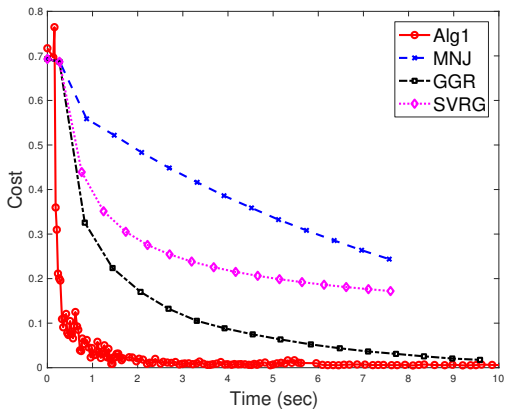
Ongoing work – scaling up to large problems

Key innovations:

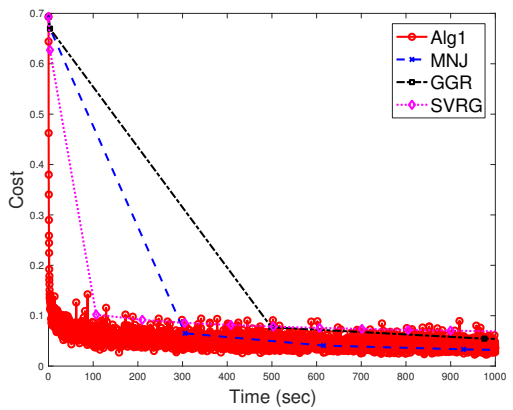
- Replace the GP with a matrix updated using fast Cholesky routines.
- Exploit a receding history of iterates and gradients akin to L-BFGS.
- An auxiliary variable Markov chain construction.



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.

Adrian Wills and Thomas B. Schön. **Stochastic quasi-Newton with adaptive step lengths for large-scale problems**. *arXiv:1802.04310*, February, 2018.

Conclusions

Derived a **probabilistic** quasi-Newton algorithm that can be used with **noisy** observations of the cost function and its derivatives.

- Non-standard interpretation of quasi-Newton.
- Represent the Hessian using a Gaussian process.
- Application: Maximum likelihood estimation in nonlinear SSMs.
- We can scale up to large problems.

Adrian G. Wills and Thomas B. Schön. **On the construction of probabilistic Newton-type algorithms**, *Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, Melbourne, Australia, December 2017.

Significantly updated material will soon be available.