Derivative Free Optimization class
AMS & Optimization Masters
On the connection between affine-invariance, convergence and learning second order information

Anne Auger
Inria and CMAP, Ecole Polytechnique
France

Motivation

CMA-ES: a widely used randomized DFO algorithm [Hansen et al. 2001-2006]

for non-convex, non-smooth, difficult black-box problems

parameter-free

6.5 + 54 millions downloads for two main Python codes

does not even use function value

yet observed to learn "second-order" information in particular on $f(x) = g((x - x^\star)^\top H(x - x^\star))$, $H \succ 0$, $g : \mathbb{R} \to \mathbb{R}$ strict. increasing

sometimes presented as (randomized) quasi-Newton

How is that even possible??

Objectives

uncover the simple and nice mathematical arguments behind this learning

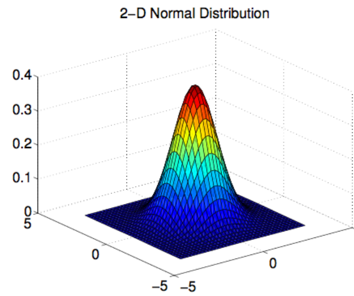
→ illustrate proofs on quasi-Newton algorithms (BFGS)

simpler and illustrates the generality of the ideas

Disclaimer: results on quasi-Newton presented are not new nor impressive *stronger results exists*

yet we show that they stem from simple fundamental properties

Adaptive Stochastic Optimization Algorithm

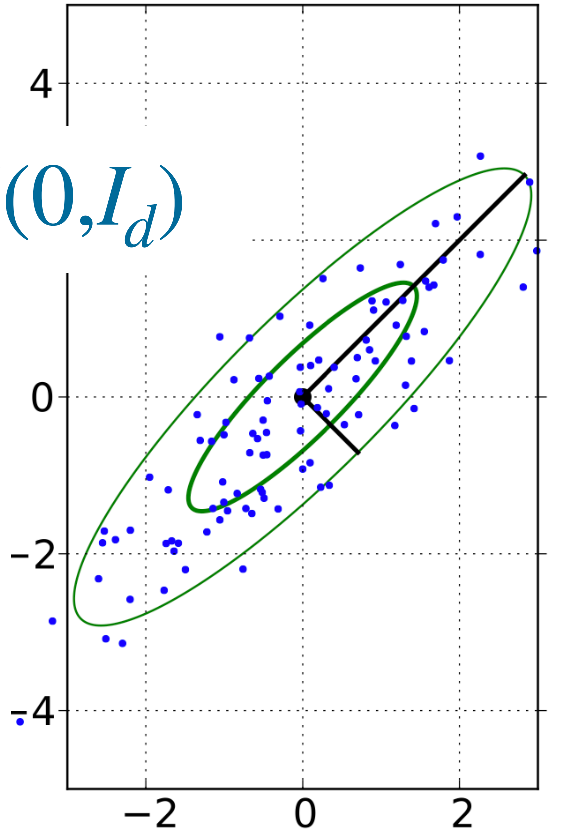


Given e.g. $\theta_t = (m_t, \sigma_t, C_t) \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

❶ Sample candidate solutions $X^i_{t+1} \sim \mathcal{N}(m_t, \sigma_t^2 C_t)$, i.e.

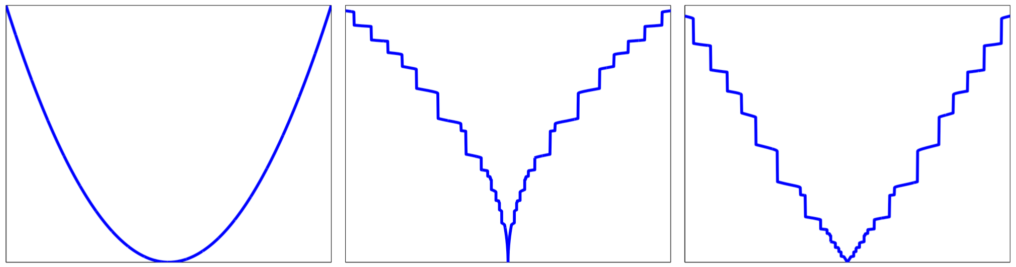
$$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t} U^i_{t+1}, \; i = 1, \ldots, \lambda$$

$\{U_t, t \geq 1\}$ i.i.d.,$U^i_{t+1} \sim \mathcal{N}(0, I_d)$



❷ Evaluate and rank candidate solutions

$$f\left(X^{s_{t+1}(1)}_{t+1}\right) \leq \ldots \leq f\left(X^{s_{t+1}(\lambda)}_{t+1}\right)$$



❸ Update θ_t:

$$\theta_{t+1} = F\left(\theta_t, [U^{s_{t+1}(1)}_{t+1}, \ldots, U^{s_{t+1}(\lambda)}_{t+1}]\right)$$

should drive m_t towards the optimum

CMA-ES - simplified setting $\theta_t = (m_t, \sigma_t, C_t) \ \in \mathbb{R}^n \times \mathbb{R}_> \times \mathcal{S}^n_{++}$

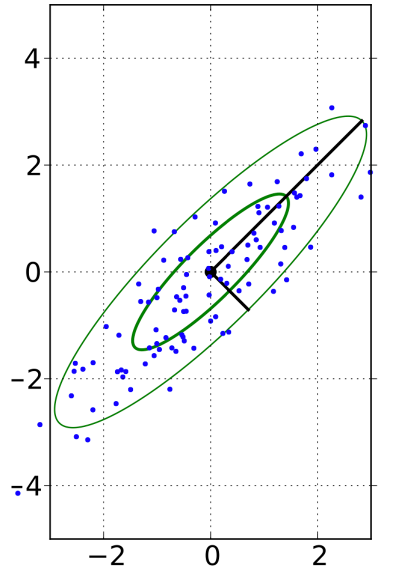
Sampling + ranking:

$y_i \sim \mathcal{N}(0, C_t)$

$$X^i_{t+1} = m_t + \sigma_t \sqrt{C_t} U^i_{t+1} \ \ i = 1, \ldots, \lambda$$

$$f\left(X^{s_{t+1}(1)}_{t+1}\right) \leq \ldots \leq f\left(X^{s_{t+1}(\lambda)}_{t+1}\right)$$

$\{U_t, t \geq 1\}$ i.i.d $U^i_{t+1} \sim \mathcal{N}(0, I_d)$



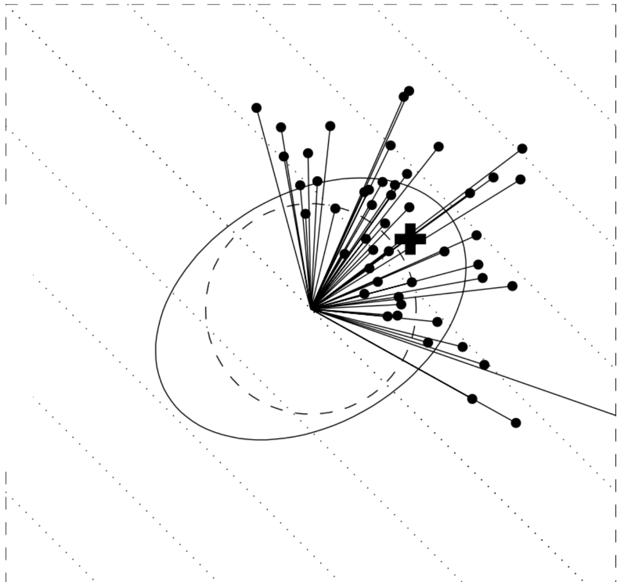
Update of θ_t:

$$m_{t+1} = \sum_{i=1}^{\mu} w_i X^{s_{t+1}(i)}_{t+1} = m_t + \sigma_t \sqrt{C_t} \sum_{i=1}^{\mu} w_i U^{s_{t+1}(i)}_{t+1}$$

$\sum_{i=1}^{\mu} w_i = 1, \mu_{\text{eff}} = 1 / \sum w_i^2$

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{c_\sigma}{d_\sigma} \left[\frac{\sqrt{\mu_{\text{eff}}} \| \sum_{i=1}^{\mu} w_i U^{s_{t+1}(i)}_{t+1} \|}{E[\| \mathcal{N}(0, I_d) \|]} - 1 \right] \right)$$

$$C_{t+1} = (1 - c_\mu \qquad) C_t + \underbrace{c_\mu \sqrt{C_t} \left(\sum_{i=1}^{\mu} w_i U^{s_{t+1}(i)}_{t+1} [U^{s_{t+1}(i)}_{t+1}]^\top \right) \sqrt{C_t}}_{\textit{rank } \mu \textit{ update}}$$

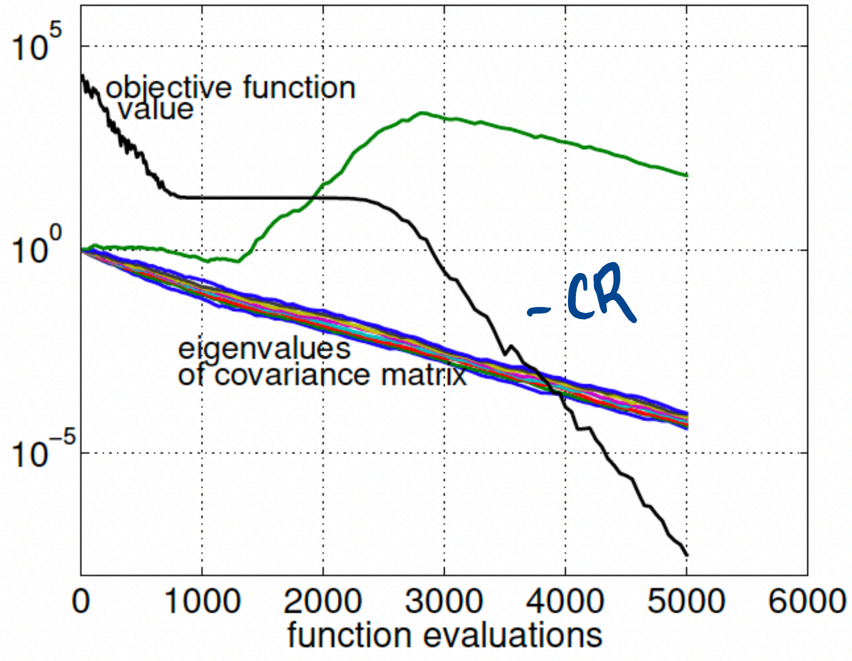


CMA-ES: Linear Convergence and Learning Inverse Hessian

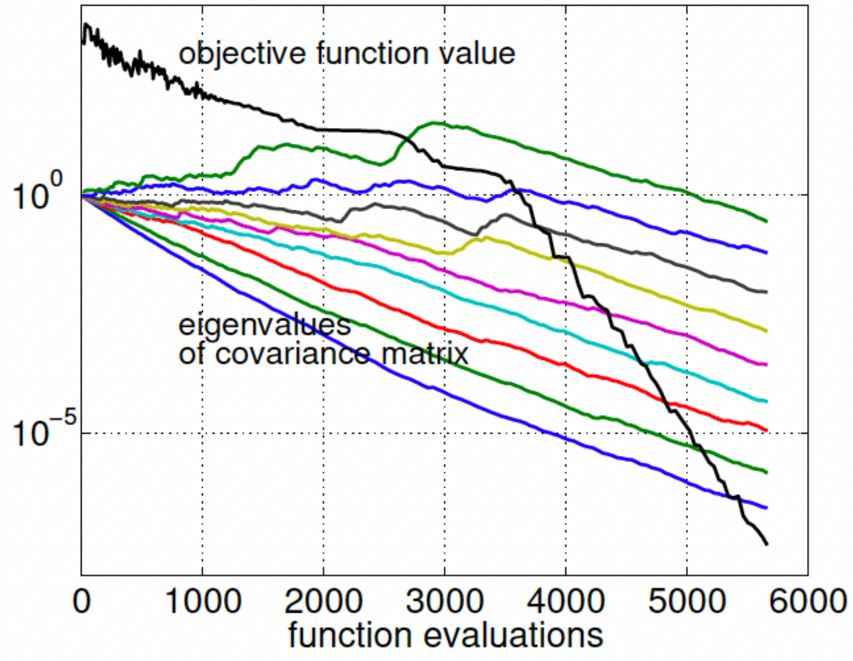
For all $f(x) = g\left(\frac{1}{2}(x - x^\star)^\top H (x - x^\star)\right)$, with $g : \mathrm{Im}(f) \to \mathbb{R}$ strict increasing, $H \succ 0$ (SDP)

$$\frac{1}{t} \ln \frac{\|m_t - x^\star\|}{\|m_0 - x^\star\|} \xrightarrow[t\to\infty]{} -\mathrm{CR} \qquad\qquad C_t \propto \alpha_t H^{-1} \quad \text{with } \alpha_t \to 0$$

Empirical observations:

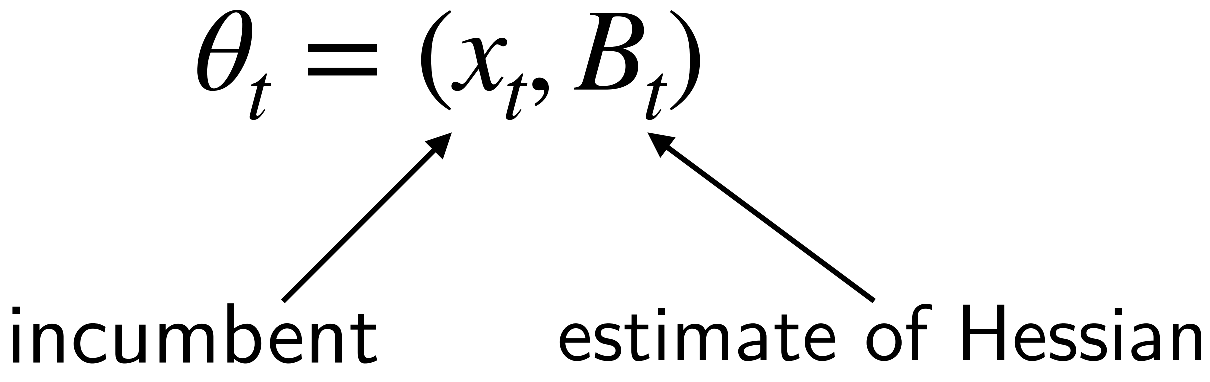


$$\mathrm{Eig}(H) = \left(10^{-6}, 1, \dots, 1\right)$$



$$\mathrm{Eig}(H) = \left(1, \dots, 10^{6\frac{i-1}{n-1}}, \dots, 10^6\right)$$

BFGS algorithm



$$\theta_t = (x_t, B_t)$$

incumbent

estimate of Hessian

```
initialize state $\theta_0 = (x_0, B_0) \in \mathbb{R}^n \times \mathscr{S}(n, \mathbb{R})$, $t = 0$
while stopping criterion not met do
    compute $p_t = -B_t^{-1} \nabla f(x_t)$
    compute step-size: $\alpha_t = \text{LineSearch}(x_t, p_t, f)$
    move in the direction of $p_t$: $x_{t+1} = x_t + \alpha_t p_t = x_t - \alpha_t B_t^{-1} \nabla f(x_t)$
    compute $s_t = \alpha_t p_t$
    compute $y_t = \nabla f(x_{t+1}) - \nabla f(x_t)$
    update estimate of Hessian: $B_{t+1} = B_t + \frac{y_t y_t^\mathrm{T}}{y_t^\mathrm{T} s_t} - \frac{B_t s_t s_t^\mathrm{T} B_t}{s_t^\mathrm{T} B_t s_t}$
    $t = t + 1$
end while
```

On affine-invariance

Affine-invariance

on $x \mapsto f(x)$ x_0 x_1 x_2 $\ldots$ x_t $\ldots$

Affine-invariance

$A \in \mathrm{GL}n(\mathbb{R})$

on $x \mapsto f(x)$ x_0 x_1 x_2 $\dots$ x_t $\dots$

on $x' \mapsto f(Ax')$ x'_0 x'_1 x'_2 $\dots$ x'_t $\dots$

$f(Ax'+\cancel{b})$

↑ because of translation invariance

Affine-invariance

$A \in \mathrm{GL}n(\mathbb{R})$

$$\begin{tikzcd}
\text{on } x \mapsto f(x) & x_0 \arrow[d, leftrightarrow] & x_1 \arrow[d, leftrightarrow] & x_2 \arrow[d, leftrightarrow] & \dots & x_t \arrow[d, leftrightarrow] & \dots \\
\text{on } x' \mapsto f(Ax') & x'_0 & x'_1 & x'_2 & \dots & x'_t & \dots \\
 & \parallel & \parallel & \parallel & & \parallel & \\
 & A^{-1}x_0 & A^{-1}x_1 & A^{-1}x_2 & \dots & A^{-1}x_t & \dots
\end{tikzcd}$$

Affine-invariance

An algorithm is affine invariant if it produces the same trajectory when optimizing $f(x)$ or any $f(Ax')$ with $A \in \mathrm{GL}n(\mathbb{R})$

up to the change of variable: $x' = A^{-1}x$

$$\begin{tikzcd}
\text{on } x \mapsto f(x) & x_0 \arrow[d, leftrightarrow] & x_1 \arrow[d, leftrightarrow] & x_2 \arrow[d, leftrightarrow] & \dots & x_t \arrow[d, leftrightarrow] & \dots \\
\text{on } x' \mapsto f(Ax') & x'_0 \arrow[d, equal] & x'_1 \arrow[d, equal] & x'_2 \arrow[d, equal] & \dots & x'_t \arrow[d, equal] & \dots \\
 & A^{-1}x_0 \arrow[d, equal] & A^{-1}x_1 & A^{-1}x_2 & \dots & A^{-1}x_t & \dots \\
 & \Phi_A(x_0) & & & & &
\end{tikzcd}$$

Affine-invariance: commutative diagram

An algorithm is affine-invariant if for all $A \in \mathrm{GL}n(\mathbb{R})$, there exists Φ_A (a bijective change of state variables) s.t. the diagram commutes:

change of state variables

$$\begin{tikzcd}
x_t \arrow[r, "x \mapsto f(x)"] \arrow[d, shift right, "\Phi_A"'] & x_{t+1} \arrow[d, shift right, "\Phi_A"'] \\
x'_t \arrow[r, "x' \mapsto f(Ax')"] \arrow[u, shift right, "\Phi_{A^{-1}}"'] & x'_{t+1} \arrow[u, shift right, "\Phi_{A^{-1}}"']
\end{tikzcd}$$

$$x'_t = \Phi_A(x_t) = A^{-1} x_t$$

Affine-invariance: commutative diagram

often state not reduced to incumbent solutions

example: BFGS where $\theta_t = (x_t, B_t)$

Affine-invariance: commutative diagram

often state not reduced to incumbent solutions

example: BFGS where $\theta_t = (x_t, B_t)$

An algorithm is affine-invariant if for all $A \in \mathrm{GL}n(\mathbb{R})$, there exists a bijective change of state variables Φ_A s.t. the diagram commutes:

change of state variables

$$\begin{tikzcd}
\theta_t \arrow[r, "x \mapsto f(x)"] \arrow[d, shift right, "\Phi_A"'] & \theta_{t+1} \arrow[d, shift right, "\Phi_A"'] \\
\theta'_t \arrow[r, "x' \mapsto f(Ax')"] \arrow[u, shift right, "\Phi_{A^{-1}}"'] & \theta'_{t+1} \arrow[u, shift right, "\Phi_{A^{-1}}"']
\end{tikzcd}$$

$$\theta'_t = \Phi_A(\theta_t)$$

$$\theta_t = \Phi_{A^{-1}}(\theta'_t)$$

Affine-invariance: commutative diagram

Affine-invariance ⇒ rotational invariance

An algorithm is affine-invariant if for all $A \in \mathrm{GL}n(\mathbb{R})$, there exists a bijective change of state variables Φ_A s.t. the diagram commutes:

change of state variables

$$\begin{tikzcd}
\theta_t \arrow[r, "x \mapsto f(x)"] \arrow[d, shift right, "\Phi_A"'] & \theta_{t+1} \arrow[d, shift right, "\Phi_A"'] \\
\theta'_t \arrow[r, "x' \mapsto f(Ax')"] \arrow[u, shift right, "\Phi_{A^{-1}}"'] & \theta'_{t+1} \arrow[u, shift right, "\Phi_{A^{-1}}"']
\end{tikzcd}$$

$$\theta'_t = \Phi_A(\theta_t)$$

$$\theta_t = \Phi_{A^{-1}}(\theta'_t)$$

Affine-invariance of BFGS

The BFGS algorithm (with affine-invariant step-size) satisfies for all $A \in GL(\mathbb{R}^n)$ the commutative diagram:

affine-invariant step-size: constant, exact line-search, …

$$\begin{tikzcd}
(x_t, B_t) \arrow[rr, "x \mapsto f(x)"] \arrow[d, shift right, "\Phi_A"'] & & (x_{t+1}, B_{t+1}) \arrow[d, shift right, "\Phi_A"'] \\
(x'_t, B'_t) \arrow[rr, "x' \mapsto f(Ax')"] \arrow[u, shift right, "\Phi_{A^{-1}}"'] & & (x'_{t+1}, B'_{t+1}) \arrow[u, shift right, "\Phi_{A^{-1}}"']
\end{tikzcd}$$

with $(x'_t, B'_t) = \Phi_A(x_t, B_t) := (A^{-1}x_t, A^\top B_t A)$

Exercice: prove that the BFGS algorithm is affine-invariant

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a Frechet differentiable objective function. Consider the BFGS algorithm defined as

```
initialize state $\theta_0 = (x_0, B_0) \in \mathbb{R}^n \times \mathcal{S}_{n,>}(\mathbb{R})$, $k = 0$
while stopping criterion not met do
    compute $d_k = -B_k^{-1} \nabla f(x_k)$
    compute step-size: $\alpha_k = \mathrm{LineSearch}(x_k, d_k, f)$
    move in the direction of $d_k$: $x_{k+1} = x_k + \alpha_k d_k$
    compute $s_k = \alpha_k d_k$
    compute $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
    update estimate of Hessian: $B_{k+1} = B_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k}$
    $k = k + 1$
end while
```

We will assume for the sake of simplicity that the step-size $\alpha_k = \alpha$ is constant.

Let $A \in \mathbb{R}^{n\times n}$ be an invertible matrix and let $x_0 \in \mathbb{R}^n$ and $B_0 \in \mathbb{R}^{n\times n}$ with $B_0 \succ 0$. Consider the sequence $(x_k, B_k)_{k\geq 1}$ generated by the BFGS algorithm optimizing $x \mapsto f(x)$. Let $(x'_0, B'_0) = (A^{-1}x_0, A^T B_0 A)$ and consider $(x'_k, B'_k)_{k\geq 1}$ the sequence of states of the BFGS algorithm optimizing $g(x') = f(Ax')$ and initialized in (x'_0, B'_0).

Prove that for all $k \geq 1$, $(x'_k, B'_k) = (A^{-1}x_k, A^T B_k A)$, i.e. that the BFGS algorithm is affine-invariant.