Quasi-Newton methods

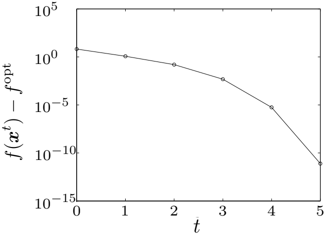
Ye Shi

ShanghaiTech University

Newton's method

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x})$$

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - (\nabla^2 f(\boldsymbol{x}^t))^{-1}\nabla f(\boldsymbol{x}^t)$$



- quadratic convergence: attains ε accuracy within $O(\log\log\frac{1}{\varepsilon})$ iterations
- typically requires storing and inverting Hessian $\nabla^2 f(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$
- a single iteration may last forever; prohibitive storage requirement

Quasi-Newton methods

key idea: approximate the Hessian matrix using only gradient information

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \underbrace{\boldsymbol{H}_t}_{\text{surrogate of } (\nabla^2 f(\boldsymbol{x}^t))^{-1}} \nabla f(\boldsymbol{x}^t)$$

challenges: how to find a good approximation $\boldsymbol{H}_t \succ \boldsymbol{0}$ of $\left(\nabla^2 f(\boldsymbol{x}^t)\right)^{-1}$

- using only gradient information
- using limited memory
- achieving super-linear convergence

Criterion for choosing $\boldsymbol{H}_t$

Consider the following *approximate quadratic model* of $f(\cdot)$:

$$f_t(\boldsymbol{x}) := f(\boldsymbol{x}^{t+1}) + \langle \nabla f(\boldsymbol{x}^{t+1}), \boldsymbol{x} - \boldsymbol{x}^{t+1} \rangle + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{t+1})^\top \boldsymbol{H}_{t+1}^{-1} (\boldsymbol{x} - \boldsymbol{x}^{t+1})$$

which satisfies

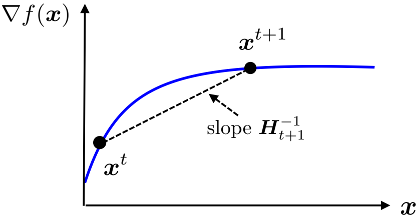
$$\nabla f_t(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^{t+1}) + \boldsymbol{H}_{t+1}^{-1}(\boldsymbol{x} - \boldsymbol{x}^{t+1})$$

One reasonable criterion: gradient matching for the latest two iterates:

$$\nabla f_t(\boldsymbol{x}^t) = \nabla f(\boldsymbol{x}^t) \tag{13.1a}$$

$$\nabla f_t(\boldsymbol{x}^{t+1}) = \nabla f(\boldsymbol{x}^{t+1}) \tag{13.1b}$$

Secant equation



(13.1b) holds automatically. To satisfy (13.1a), one requires

$$\nabla f(\boldsymbol{x}^{t+1}) + \boldsymbol{H}_{t+1}^{-1}(\boldsymbol{x}^t - \boldsymbol{x}^{t+1}) = \nabla f(\boldsymbol{x}^t)$$

$$\iff \qquad \underbrace{\boldsymbol{H}_{t+1}^{-1}(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t) = \nabla f(\boldsymbol{x}^{t+1}) - \nabla f(\boldsymbol{x}^t)}_{\text{secant equation}}$$

- the secant equation requires that $\boldsymbol{H}_{t+1}^{-1}$ maps the displacement $\boldsymbol{x}^{t+1} - \boldsymbol{x}^t$ into the change of gradients $\nabla f(\boldsymbol{x}^{t+1}) - \nabla f(\boldsymbol{x}^t)$

Secant equation

$$\boldsymbol{H}_{t+1} \underbrace{\left(\nabla f(\boldsymbol{x}^{t+1}) - \nabla f(\boldsymbol{x}^{t})\right)}_{=: \boldsymbol{y}_t} = \underbrace{\boldsymbol{x}^{t+1} - \boldsymbol{x}^{t}}_{=: \boldsymbol{s}_t} \qquad (13.2)$$

- only possible when $\boldsymbol{s}_t^\top \boldsymbol{y}_t > 0$, since

$$\boldsymbol{s}_t^\top \boldsymbol{y}_t = \boldsymbol{y}_t^\top \boldsymbol{H}_{t+1} \boldsymbol{y}_t > 0$$

- admit an infinite number of solutions, since the degrees of freedom $O(n^2)$ in choosing $\boldsymbol{H}_{t+1}^{-1}$ far exceeds the number of constraints n in (13.2)
- which $\boldsymbol{H}_{t+1}^{-1}$ shall we choose?

Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

Broyden, Fletcher, Goldfarb, Shanno

Closeness to $\boldsymbol{H}_t$

In addition to the secant equation, choose $\boldsymbol{H}_{t+1}$ sufficiently close to $\boldsymbol{H}_t$:

$$
\begin{aligned}
\text{minimize}_{\boldsymbol{H}} \quad & \|\boldsymbol{H} - \boldsymbol{H}_t\| \\
\text{subject to} \quad & \boldsymbol{H} = \boldsymbol{H}^\top \\
& \boldsymbol{H}\boldsymbol{y}_t = \boldsymbol{s}_t
\end{aligned}
$$

for some norm $\|\cdot\|$

- exploit past information regarding $\boldsymbol{H}_t$
- choosing different norms $\|\cdot\|$ results in different quasi-Newton methods

Choice of norm in BFGS

Choosing $\|\boldsymbol{M}\| := \|\boldsymbol{W}^{1/2}\boldsymbol{M}\boldsymbol{W}^{1/2}\|_{\mathrm{F}}$ for *any* weight matrix $\boldsymbol{W}$ obeying $\boldsymbol{W}\boldsymbol{s}_t = \boldsymbol{y}_t$, we get

$$\begin{aligned} \text{minimize}_{\boldsymbol{H}} \quad & \|\boldsymbol{W}^{1/2}(\boldsymbol{H} - \boldsymbol{H}_t)\boldsymbol{W}^{1/2}\|_{\mathrm{F}} \\ \text{subject to} \quad & \boldsymbol{H} = \boldsymbol{H}^\top \\ & \boldsymbol{H}\boldsymbol{y}_t = \boldsymbol{s}_t \end{aligned}$$

This admits a closed-form expression

$$\underbrace{\boldsymbol{H}_{t+1} = (\boldsymbol{I} - \rho_t \boldsymbol{s}_t \boldsymbol{y}_t^\top)\boldsymbol{H}_t(\boldsymbol{I} - \rho_t \boldsymbol{y}_t \boldsymbol{s}_t^\top) + \rho_t \boldsymbol{s}_t \boldsymbol{s}_t^\top}_{\text{BFGS update rule;} \quad \boldsymbol{H}_{t+1} \succeq \boldsymbol{0} \text{ if } \boldsymbol{H}_t \succeq \boldsymbol{0}} \tag{13.3}$$

with $\rho_t = \frac{1}{\boldsymbol{y}_t^\top \boldsymbol{s}_t}$

An alternative interpretation

$\boldsymbol{H}_{t+1}$ is also the solution to

$$\begin{aligned} \text{minimize}_{\boldsymbol{H}} \quad & \underbrace{\langle \boldsymbol{H}_t, \boldsymbol{H}^{-1} \rangle - \log\det\left(\boldsymbol{H}_t \boldsymbol{H}^{-1}\right) - n}_{\text{KL divergence between } \mathcal{N}(\boldsymbol{0}, \boldsymbol{H}^{-1}) \text{ and } \mathcal{N}(\boldsymbol{0}, \boldsymbol{H}_t^{-1})} \\ \text{subject to} \quad & \boldsymbol{H}\boldsymbol{y}_t = \boldsymbol{s}_t \end{aligned}$$

- minimizing some sort of KL divergence subject to the secant equation constraints

BFGS methods

Algorithm 13.1 BFGS

1: **for** $t = 0, 1, \cdots$ **do**

2: $\quad \boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta_t \boldsymbol{H}_t \nabla f(\boldsymbol{x}^t)$ (line search to determine η_t)

3: $\quad \boldsymbol{H}_{t+1} = (\boldsymbol{I} - \rho_t \boldsymbol{s}_t \boldsymbol{y}_t^\top) \boldsymbol{H}_t (\boldsymbol{I} - \rho_t \boldsymbol{y}_t \boldsymbol{s}_t^\top) + \rho_t \boldsymbol{s}_t \boldsymbol{s}_t^\top$, where $\boldsymbol{s}_t = \boldsymbol{x}^{t+1} - \boldsymbol{x}^t$, $\boldsymbol{y}_t = \nabla f(\boldsymbol{x}^{t+1}) - \nabla f(\boldsymbol{x}^t)$, and $\rho_t = \frac{1}{\boldsymbol{y}_t^\top \boldsymbol{s}_t}$

- each iteration costs $O(n^2)$ (in addition to computing gradients)
- no need to solve linear systems or invert matrices
- *no magic formula for initialization;* possible choices: approximate inverse Hessian at $\boldsymbol{x}^0$, or identity matrix

Rank-2 update on $\boldsymbol{H}_t^{-1}$

From the Sherman-Morrison-Woodbury formula $\left(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{V}^\top\right)^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{I}+\boldsymbol{V}^\top\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}^\top\boldsymbol{A}^{-1}$, we can show that the BFGS rule is equivalent to

$$\underbrace{\boldsymbol{H}_{t+1}^{-1}=\boldsymbol{H}_t^{-1}-\frac{1}{\boldsymbol{s}_t^\top\boldsymbol{H}_t^{-1}\boldsymbol{s}_t}\boldsymbol{H}_t^{-1}\boldsymbol{s}_t\boldsymbol{s}_t^\top\boldsymbol{H}_t^{-1}+\rho_t\boldsymbol{y}_t\boldsymbol{y}_t^\top}_{\text{rank-2 update}}$$

Local superlinear convergence

Theorem 13.1 (informal)

Suppose f is strongly convex and has Lipschitz-continuous Hessian. Under mild conditions, BFGS achieves

$$\lim_{t\to\infty} \frac{\left\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^*\right\|_2}{\left\|\boldsymbol{x}^{t}-\boldsymbol{x}^*\right\|_2} = 0$$

- *iteration complexity:* larger than Newton methods but smaller than gradient methods
- *asymptotic result:* holds when $t \to \infty$

Key observation

The BFGS update rule achieves

$$\lim_{t\to\infty} \frac{\left\|\left(\boldsymbol{H}_t^{-1} - \nabla^2 f(\boldsymbol{x}^*)\right)\left(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\right)\right\|_2}{\left\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\right\|_2} = 0$$

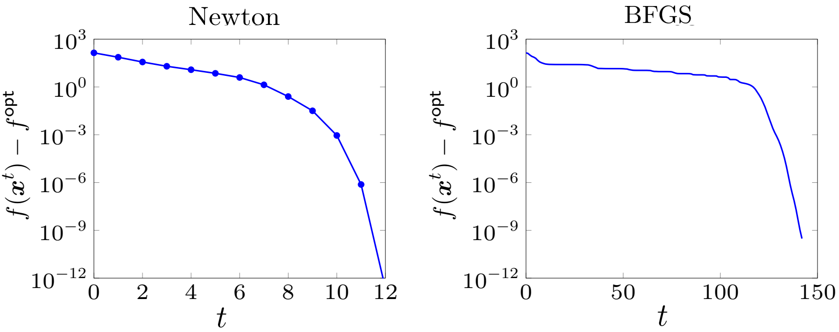
Implications

- even though $\boldsymbol{H}_t^{-1}$ may not converge to $\nabla^2 f(\boldsymbol{x}^*)$, it becomes an increasingly more accurate approximation of $\nabla^2 f(\boldsymbol{x}^*)$ along the search direction $\boldsymbol{x}^{t+1} - \boldsymbol{x}^t$
- asymptotically, $\boldsymbol{x}^{t+1} - \boldsymbol{x}^t \approx \underbrace{-\left(\nabla^2 f(\boldsymbol{x}^t)\right)^{-1}\nabla f(\boldsymbol{x}^t)}_{\text{Newton search direction}}$

Numerical example

— EE236C lecture notes

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{x} - \sum_{i=1}^{N} \log\left(b_i - \boldsymbol{a}_i^\top \boldsymbol{x}\right)$$



$$n = 100, \;\; N = 500$$

Limited-memory quasi-Newton methods

Hessian matrices are usually dense. For large-scale problems, even storing the (inverse) Hessian matrices is prohibitive

Instead of storing full Hessian approximations, one may want to maintain more parsimonious approximation of the Hessians, using only a few vectors

Limited-memory BFGS (L-BFGS)

$$\underbrace{\boldsymbol{H}_{t+1} = \boldsymbol{V}_t^\top \boldsymbol{H}_t \boldsymbol{V}_t + \rho_t \boldsymbol{s}_t \boldsymbol{s}_t^\top}_{\text{BFGS update rule}} \quad \text{with } \boldsymbol{V}_t = \boldsymbol{I} - \rho_t \boldsymbol{y}_t \boldsymbol{s}_t^\top$$

key idea: maintain a modified version of $\boldsymbol{H}_t$ *implicitly* by storing m (e.g. 20) most recent vector pairs $(\boldsymbol{s}_t, \boldsymbol{y}_t)$

Limited-memory BFGS (L-BFGS)

L-BFGS maintains

$$\begin{aligned}
\boldsymbol{H}_t^{\mathsf{L}} = {} & \boldsymbol{V}_{t-1}^\top \cdots \boldsymbol{V}_{t-m}^\top \boldsymbol{H}_{t,0}^{\mathsf{L}} \boldsymbol{V}_{t-m} \cdots \boldsymbol{V}_{t-1} \\
& + \rho_{t-m} \boldsymbol{V}_{t-1}^\top \cdots \boldsymbol{V}_{t-m+1}^\top \boldsymbol{s}_{t-m} \boldsymbol{s}_{t-m}^\top \boldsymbol{V}_{t-m+1} \cdots \boldsymbol{V}_{t-1} \\
& + \rho_{t-m+1} \boldsymbol{V}_{t-1}^\top \cdots \boldsymbol{V}_{t-m+2}^\top \boldsymbol{s}_{t-m+1} \boldsymbol{s}_{t-m+1}^\top \boldsymbol{V}_{t-m+1} \cdots \boldsymbol{V}_{t-1} \\
& + \cdots + \rho_{t-1} \boldsymbol{s}_{t-1} \boldsymbol{s}_{t-1}^\top
\end{aligned}$$

- can be computed recursively
- initialization $\boldsymbol{H}_{t,0}^{\mathsf{L}}$ may vary from iteration to iteration
- only needs to store $\{(\boldsymbol{s}_i, \boldsymbol{y}_i)\}_{t-m \le i < t}$

Reference

[1] "*Numerical optimization*, J. Nocedal, S. Wright, 2000.

[2] "*Optimization methods for large-scale systems, EE236C lecture notes*," L. Vandenberghe, UCLA.

[3] "*Optimization methods for large-scale machine learning*," L. Bottou et al., arXiv, 2016.

[4] "*Convex optimization, EE364B lecture notes*," S. Boyd, Stanford.