EECS 559 Optimization Methods for SIPML

Lecture 4 – Smooth Unconstrained Optimization

Instructor: Prof. Qing Qu (qingqu@umich.edu)

Lecture Agenda

- **Basics of Iterative Methods**
- (Accelerated) Gradient Descent for Smooth Problems
- Linesearch Methods for Step Size
- Exploiting Function Properties for Faster Convergence
- Newton & Quasi-Newton Method

Smooth Unconstrained Minimization

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathcal{C}.$$

- Suppose $f(\boldsymbol{x})$ is continuously differentiable over $\mathcal{C}$
- Consider unconstrained problem $\mathcal{C} = \mathbb{R}^n$
- Solve the problem via *iterative optimization methods,* producing a sequence of points

$$\boldsymbol{x}_1, \ \boldsymbol{x}_2, \ \cdots, \boldsymbol{x}_k, \cdots$$

starting from an initialization $\boldsymbol{x}_0$.

Smooth Unconstrained Minimization

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Goal: generate a sequence $\{\boldsymbol{x}_k\}_{k\geq 1}$ from an initialization $\boldsymbol{x}_0$, such that it quickly converges to a minimizer $\boldsymbol{x}_\star$ of f.

$$\boxed{\text{complexity} \ = \ \text{per iter cost} \ \times \ \#\text{of iterations.}}$$

Smooth Unconstrained Minimization

$$\boxed{\text{complexity} \ = \ \text{per iter cost} \ \times \ \#\text{of iterations.}}$$

- **Per iteration cost.** How much computation it takes to generate the next point $\boldsymbol{x}_{k+1}$ from $\boldsymbol{x}_k$;
- **Convergence rate.** How quickly the sequence $\{\boldsymbol{x}_k\}_{k\geq 1}$ converges, measured by

Distance to a minimizer	$\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2$
Sub-optimality in objective	$\|f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star)\|$
Gradient	$\|\nabla f(\boldsymbol{x}_k)\|_2$

Iterative Descent Algorithms

Starting with an initialization $\boldsymbol{x}_0$, construct a sequence $\{\boldsymbol{x}_k\}_{k\geq 1}$ such that

$$f(\boldsymbol{x}_{k+1}) \ < \ f(\boldsymbol{x}_k), \quad k \ = \ 0, 1, \cdots$$

- In each iteration, search in a **descent direction**:

$$\boldsymbol{x}_{k+1} \ = \ \boldsymbol{x}_k \ + \ \underbrace{\tau_k}_{\textbf{stepsize}} \ \cdot \ \underbrace{\boldsymbol{d}_k}_{\textbf{descent direction}}.$$

- The direction $\boldsymbol{d}$ is said to be a descent direction at $\boldsymbol{x}$ if

$$f'(\boldsymbol{x}; \boldsymbol{d}) \ := \ \underbrace{\lim_{\tau \searrow 0} \frac{f(\boldsymbol{x} + \tau \boldsymbol{d}) - f(\boldsymbol{x})}{\tau}}_{\text{direction derivative}} \ = \ \nabla f(\boldsymbol{x})^\top \boldsymbol{d} \ < \ 0.$$

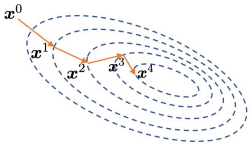
Lecture Agenda

- Basics of Iterative Methods
- **(Accelerated) Gradient Descent for Smooth Problems**
- Linesearch Methods for Step Size
- Exploiting Function Properties for Faster Convergence
- Newton & Quasi-Newton Method

Gradient Descent (GD)

One of the most important examples: **gradient descent**

$$\boldsymbol{x}_{k+1} \ = \ \boldsymbol{x}_k \ - \ \tau_k \cdot \nabla f(\boldsymbol{x}_k)$$



- can be traced back to Augustin Louis Cauchy'1847 ...

Gradient Descent (GD)

One of the most important examples: **gradient descent**

$$\boxed{\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k \;-\; \tau_k \cdot \nabla f(\boldsymbol{x}_k).}$$

- Descent direction: $\boldsymbol{d}_k \;=\; -\nabla f(\boldsymbol{x}_k)$
- a.k.a., *steepest descent,* since from Cauchy-Schwarz inequality

$$\arg\min_{\|\boldsymbol{d}\|_2 \le 1} f'(\boldsymbol{x};\boldsymbol{d}) \;=\; \arg\min_{\|\boldsymbol{d}\|_2 \le 1} \nabla f(\boldsymbol{x})^\top \boldsymbol{d} \;=\; -\nabla f(\boldsymbol{x}) / \|\nabla f(\boldsymbol{x})\|_2 .$$

Gradient Descent (GD)

One of the most important examples: **gradient descent**

$$\boxed{\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k \;-\; \tau_k \cdot \nabla f(\boldsymbol{x}_k).}$$

- the scalar $\tau_k > 0$ is the *step size*:
  - ➢ can be determined *analytically* from the property of the function f
  - ➢ *numerically* computed via a *line search*:

$$\tau_k \;=\; \arg\min_{t \ge 0} \; f(\boldsymbol{x}_k - t \cdot \nabla f(\boldsymbol{x}_k))$$

Convergence of GD

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \; f(\boldsymbol{x})$$

Assumptions:

- $f : \mathbb{R}^n \mapsto \mathbb{R}$ is **convex** and **smooth**;
- The gradient $\nabla f(\boldsymbol{x})$ is L-**Lipschitz** with

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}')\|_2 \;\le\; L \cdot \|\boldsymbol{x} - \boldsymbol{x}'\|_2 , \quad \forall\, \boldsymbol{x},\, \boldsymbol{x}' \in \mathbb{R}^n.$$

Convergence of GD

Lemma. (Sufficient function value decrease of GD) With step size $\tau_k = 1/L$, the gradient descent iterates $\{\boldsymbol{x}_k\}_{k\ge 1}$ generated by

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k \;-\; \tau_k \cdot \nabla f(\boldsymbol{x}_k)$$

satisfy the following:

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) \;&\le\; f(\boldsymbol{x}_k) \;-\; \frac{1}{2L} \|\nabla f(\boldsymbol{x}_k)\|_2^2 \\ &<\; f(\boldsymbol{x}_k) \quad (\|\nabla f(\boldsymbol{x}_k)\|_2 \neq 0) \end{aligned}$$

Convergence of GD

Proof. Because ∇f is Lipschitz, by 1st-order Taylor approximation

$$f(\boldsymbol{x}') \le f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle + \frac{L}{2} \|\boldsymbol{x}' - \boldsymbol{x}\|_2^2, \quad \forall\, \boldsymbol{x}', \boldsymbol{x}$$

Now, let $\boldsymbol{x}' = \boldsymbol{x} - \tau \nabla f(\boldsymbol{x})$

$$\begin{aligned} f(\boldsymbol{x}') &\le f(\boldsymbol{x}) - \tau \|\nabla f(\boldsymbol{x})\|_2^2 + \frac{\tau^2 L}{2} \|\nabla f(\boldsymbol{x})\|_2^2 \\ &= f(\boldsymbol{x}) - \tau \left(1 - \frac{\tau L}{2}\right) \|\nabla f(\boldsymbol{x})\|_2^2 \end{aligned}$$

Convergence of GD

Proof. Because ∇f is Lipschitz, by 1st-order Taylor approximation

$$f(\boldsymbol{x}') \le f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle + \frac{L}{2} \|\boldsymbol{x}' - \boldsymbol{x}\|_2^2, \quad \forall\, \boldsymbol{x}', \boldsymbol{x}$$

Now, let $\boldsymbol{x}' = \boldsymbol{x} - \tau \nabla f(\boldsymbol{x})$

$$\begin{aligned} f(\boldsymbol{x}') &\le f(\boldsymbol{x}) - \tau \|\nabla f(\boldsymbol{x})\|_2^2 + \frac{\tau^2 L}{2} \|\nabla f(\boldsymbol{x})\|_2^2 \\ &= f(\boldsymbol{x}) - \tau \left(1 - \frac{\tau L}{2}\right) \|\nabla f(\boldsymbol{x})\|_2^2 \\ &\le f(\boldsymbol{x}) - \frac{1}{L}(1 - 1/2) \|\nabla f(\boldsymbol{x})\|_2^2 = f(\boldsymbol{x}) - \frac{1}{2L} \|\nabla f(\boldsymbol{x})\|_2^2 \end{aligned}$$

Convergence of GD

Theorem. ($O(1/k)$ **sublinear convergence of GD)** Suppose $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex and smooth with its gradient L-Lipschitz. If we run GD for k iterations with a fixed step size $\tau = 1/L$, it will yield a sequence $\{\boldsymbol{x}_k\}_{k\ge 1}$ such that

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \;\le\; \frac{L}{2k} \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 \;=\; O(\frac{1}{k}),$$

Moreover, as $k \to +\infty$, then $\boldsymbol{x}_k \to \boldsymbol{x}_\star$.

Convergence of GD

Proof. Given f is convex & smooth, and $\boldsymbol{x}_\star$ is an optimal solution

$$\begin{aligned} & f(\boldsymbol{x}_\star) \ge f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{x}_\star - \boldsymbol{x}) \\ \Leftrightarrow \quad & f(\boldsymbol{x}) \le f(\boldsymbol{x}_\star) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{x}_\star) \end{aligned}$$

By our previous lemma $\boldsymbol{x}' = \boldsymbol{x} - \frac{1}{L}\nabla f(\boldsymbol{x})$ $\Rightarrow$ $f(\boldsymbol{x}') \le f(\boldsymbol{x}) - \frac{1}{2L} \|\nabla f(\boldsymbol{x})\|_2^2$

$$\begin{aligned} & f(\boldsymbol{x}') \le f(\boldsymbol{x}_\star) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{x}_\star) - \frac{1}{2L} \|\nabla f(\boldsymbol{x})\|_2^2 \\ \Leftrightarrow \quad & f(\boldsymbol{x}') - f(\boldsymbol{x}_\star) \le \frac{L}{2} \left[\frac{2}{L} \nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{x}_\star) - \frac{1}{L^2} \|\nabla f(\boldsymbol{x})\|_2^2 \right] \end{aligned}$$

Convergence of GD

Therefore, we have

$$\begin{aligned} f(\boldsymbol{x}') - f(\boldsymbol{x}_\star) &\le \frac{L}{2}\left[\frac{2}{L}\nabla f(\boldsymbol{x})^\top(\boldsymbol{x}-\boldsymbol{x}_\star) - \frac{1}{L^2}\|\nabla f(\boldsymbol{x})\|_2^2\right] \\ &= \frac{L}{2}\left[\|\boldsymbol{x}-\boldsymbol{x}_\star\|_2^2 - \left\|\boldsymbol{x}-\frac{1}{L}\nabla f(\boldsymbol{x})-\boldsymbol{x}_\star\right\|_2^2\right] \\ &= \frac{L}{2}\left[\|\boldsymbol{x}-\boldsymbol{x}_\star\|_2^2 - \|\boldsymbol{x}'-\boldsymbol{x}_\star\|_2^2\right] \end{aligned}$$

Convergence of GD

Let $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - \frac{1}{L}\nabla f(\boldsymbol{x}_{k-1})$, then we have

$$\begin{aligned} \sum_{i=1}^k (f(\boldsymbol{x}_i) - f(\boldsymbol{x}_\star)) &\le \frac{L}{2}\sum_{i=1}^k \left[\|\boldsymbol{x}_{i-1}-\boldsymbol{x}_\star\|_2^2 - \|\boldsymbol{x}_i-\boldsymbol{x}_\star\|_2^2\right] \\ &= \frac{L}{2}\left[\|\boldsymbol{x}_0-\boldsymbol{x}_\star\|_2^2 - \|\boldsymbol{x}_k-\boldsymbol{x}_0\|_2^2\right] \le \frac{L}{2}\|\boldsymbol{x}_0-\boldsymbol{x}_\star\|_2^2 \end{aligned}$$

Therefore, given $f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \le f(\boldsymbol{x}_i) - f(\boldsymbol{x}_\star)$ for $\forall i \le k$,

$$k\,(f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star)) \le \sum_{i=1}^k (f(\boldsymbol{x}_i) - f(\boldsymbol{x}_\star)) \le \frac{L}{2}\|\boldsymbol{x}_0-\boldsymbol{x}_\star\|_2^2$$

Suboptimal Convergence Rate of GD

Suppose the iterates $\{\boldsymbol{x}_i\}_{i\ge 0}$ is generated from a *black-box model:*

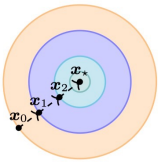
$$\boldsymbol{x}_{k+1} \;=\; \mathscr{F}_k\left(\{\boldsymbol{x}_i\}_{i=0}^k,\, \{f(\boldsymbol{x}_i)\}_{i=0}^k,\, \{\nabla f(\boldsymbol{x}_i)\}_{i=0}^k\right)$$

Theorem (Nesterov'03). For every positive L and R, there exists a convex differentiable f with ∇f L-Lipschitz, and an initial point $\boldsymbol{x}_0$ satisfying $\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2 \le R$, such that

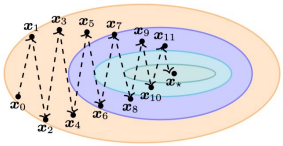
$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \;\ge\; c\frac{LR}{k^2} \;=\; \Omega(\frac{1}{k^2}),$$

where $c > 0$ is some numerical constant.

Slow Convergence of GD

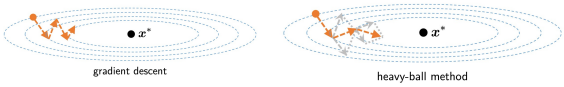


well-conditioned
(fast convergence)



Ill-conditioned
(slow convergence)

Heavy Ball Methods (Polyak'64)

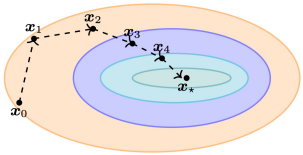


B. Polyak

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k - \tau_k\nabla f(\boldsymbol{x}_k) + \underbrace{\beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})}_{\textbf{momemtum term}}$$

- This is also called **momentum method**.
- Basis for the popular ADAM for training modern neural networks.

Heavy Ball Methods (Polyak'64)



$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \tau_k\nabla f(\boldsymbol{x}_k) \\ &\quad + \underbrace{\beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})}_{\textbf{momemtum term}} \end{aligned}$$

- This is also called **momentum method**.
- Basis for the popular ADAM for training modern neural networks.
- The worst-case convergence is still $O(1/k)$.

Nesterov's Method (Nesterov'83)

Generate two sequences $\{\boldsymbol{x}_k\}_{k\ge 1}$ and $\{\boldsymbol{p}_k\}_{k\ge 1}$

$$\begin{aligned} \boldsymbol{p}_{k+1} &= \boldsymbol{x}_k + \beta_k\cdot(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}), \\ \boldsymbol{x}_{k+1} &= \boldsymbol{p}_{k+1} - \alpha\nabla f(\boldsymbol{p}_{k+1}) \end{aligned}$$

Y. Nesterov

- Alternates between gradient updates $\boldsymbol{x}_k$ and proper extrapolation $\boldsymbol{p}_k$.
- **Not** a descent method, i.e., we may not have $f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k)$
- With properly chosen α and β_k, the method *can* achieve *optimal* convergence rate $O(1/k^2)$

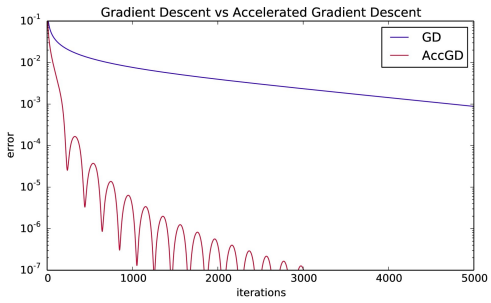
Nesterov's Method (Nesterov'83)

Theorem. (Convergence of accelerated GD) Suppose $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex and smooth with its gradient L-Lipschitz. The iterates $\{\boldsymbol{x}_k\}_{k\ge 1}$ generated by the accelerated GD method satisfy

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \;\le\; \frac{L}{2(k+1)^2}\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 \;=\; O(\frac{1}{k^2}).$$

Moreover, as $k \to +\infty$, then $\boldsymbol{x}_k \to \boldsymbol{x}_\star$.

Nesterov's Method (Nesterov'83)



Lecture Agenda

- Basics of Iterative Methods
- (Accelerated) Gradient Descent for Smooth Problems
- **Linesearch Methods for Step Size**
- Exploiting Function Properties for Faster Convergence
- Newton & Quasi-Newton Method

Exact Linesearch

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k \;-\; \tau_k \cdot \nabla f(\boldsymbol{x}_k)$$

- The choice $\tau = 1/L$ relies on the knowledge of the function property of $f(\boldsymbol{x})$
- Another more practical strategy is using linesearch

$$\tau_k \;=\; \arg\min_{t \geq 0} \; f(\boldsymbol{x}_k - t \cdot \nabla f(\boldsymbol{x}_k))$$

which could be expensive to solve...

Backtracking Linesearch

- **Armijo condition:** for some $c_1 \in (0,1)$

$$f(\boldsymbol{x}_k + \tau_k \cdot \boldsymbol{d}_k) \;<\; f(\boldsymbol{x}_k) + c_1 \cdot \tau_k \cdot \boldsymbol{d}_k^\top \nabla f(\boldsymbol{x}_k)$$

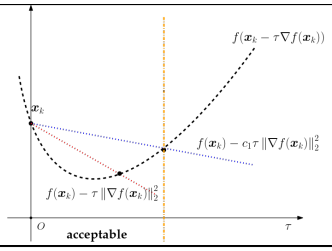
Purpose: ensure *sufficient decrease* of objective values

- **Wolfe condition:** for some $0 < c_1 < c_2 < 1$

$$f(\boldsymbol{x}_k + \tau_k \cdot \boldsymbol{d}_k) \;<\; f(\boldsymbol{x}_k) + c_1 \cdot \tau_k \cdot \boldsymbol{d}_k^\top \nabla f(\boldsymbol{x}_k)$$

$$\boldsymbol{d}_k^\top \nabla f(\boldsymbol{x}_k + \tau_k \boldsymbol{d}_k) \;\geq\; c_2 \boldsymbol{d}_k^\top \nabla f(\boldsymbol{x}_k) \quad \text{(curvature condition)}$$

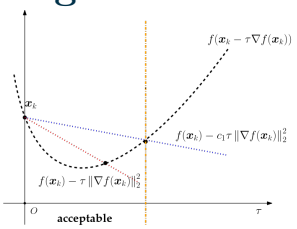
Purpose: rule out *unacceptably small* steps



Algorithm 1 Backtracking Linesearch for Gradient Descent

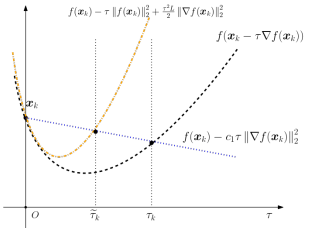
Initialize $\tau = 1$, $c_1 \in (0,1)$, $\alpha \in (0,1)$
while $f(\boldsymbol{x}_k - \tau \nabla f(\boldsymbol{x}_k)) \;>\; f(\boldsymbol{x}_k) - c_1 \tau \|\nabla f(\boldsymbol{x}_k)\|_2^2$ **do**
  $\tau \;\leftarrow\; \alpha \cdot \tau$
end while

Backtracking Linesearch for GD



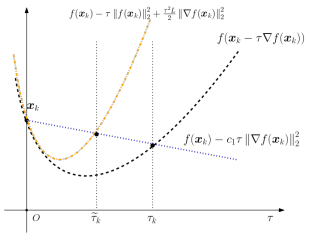
- $f(\boldsymbol{x}_k) - c_1 \tau \|\nabla f(\boldsymbol{x}_k)\|_2^2$ lies above $f(\boldsymbol{x}_k - \tau \nabla f(\boldsymbol{x}_k))$ for small τ
- ensures sufficient decrease of objective values.

Backtracking Linesearch for GD



In practice, backtracking linesearch often (but not always) provides good estimates on the local Lipschitz constants of the gradient $\nabla f(\boldsymbol{x})$.

Backtracking Linesearch for GD



Question. In training deep network, why we do *not* use linesearch for adapting the learning rate?

Backtracking Linesearch for GD

Theorem. (Boyd, Vandenberghe'04) Let the function $f(\boldsymbol{x})$ be μ-strongly convex and L-smooth. With backtracking linesearch, the GD method converges with

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \;\leq\; \left(1 - \min\left\{2c_1\mu, \frac{2\alpha c_1 \mu}{L}\right\}\right)^k (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_\star))$$

where $\boldsymbol{x}_\star$ is the unique minimizer.

Lecture Agenda

- Basics of Iterative Methods
- (Accelerated) Gradient Descent for Smooth Problems
- Linesearch Methods for Step Size
- **Exploiting Function Properties for Faster Convergence**
- Newton & Quasi-Newton Method

Assumptions of Strong Convexity (SC)

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Assumption 1. The function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is *smooth* with gradient $\nabla f(\boldsymbol{x})$ being L-Lipschitz.

Assumption 2. The function is μ*-strongly convex*. That is, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$f(\boldsymbol{y}) \;\geq\; f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$$

- Furthermore, if $f \in \mathcal{C}^2$, then $\mathbf{0} \prec \mu \boldsymbol{I} \preceq \nabla^2 f(\boldsymbol{x}) \preceq L\boldsymbol{I}$.

Linear Convergence of GD

Theorem. (Linear convergence for strong convexity) Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be smooth and μ-strongly convex, and suppose its gradient $\nabla f(\boldsymbol{x})$ is L-Lipschitz. Choose a fixed step size $\tau_k \equiv \tau = \frac{2}{\mu + L}$, then

$$\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \;\leq\; \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2,$$

where $\kappa = \frac{L}{\mu}$ is the condition number, and $\boldsymbol{x}_\star$ is the minimizer.

- A direct consequence of Lipschitzness of $\nabla f(\boldsymbol{x})$

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \;\leq\; \frac{L}{2}\left(\frac{\kappa - 1}{\kappa + 1}\right)^{2k} \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2$$

 Theorem D.4 in High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications. John Wright, Yi Ma.

Linear Convergence of GD

Proof. First, by the fundamental theorem of calculus:

$$\nabla f(\boldsymbol{x}_k) = \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_\star) = \left[\int_0^1 \nabla^2 f(\boldsymbol{x}(t))dt\right](\boldsymbol{x}_k - \boldsymbol{x}_\star)$$

where $\boldsymbol{x}(t) = (1-t)\boldsymbol{x}_k + t\boldsymbol{x}_\star$.
Then, we have

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 &= \|\boldsymbol{x}_k - \tau\nabla f(\boldsymbol{x}_k) - \boldsymbol{x}_\star\|_2 \\ &= \|(\boldsymbol{x}_k - \boldsymbol{x}_\star) - \tau(\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_\star))\|_2 \\ &= \left\|\left(\boldsymbol{I} - \tau\int_0^1 \nabla^2 f(\boldsymbol{x}(t))dt\right)(\boldsymbol{x}_k - \boldsymbol{x}_\star)\right\|_2\end{aligned}$$

Linear Convergence of GD

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 &\leq \left\|\boldsymbol{I} - \tau\int_0^1 \nabla^2 f(\boldsymbol{x}(t))dt\right\|_2 \|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \\ &\leq \left[\sup_{0\leq t\leq 1}\left\|\boldsymbol{I} - \tau\nabla^2 f(\boldsymbol{x}(t))\right\|_2\right]\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2\end{aligned}$$

By the fact that f is μ-strongly convex, $\nabla^2 f(\boldsymbol{x}(t)) \geq \mu\boldsymbol{I}$, $\forall\, t$, we have

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 &\leq (1 - \tau\mu)\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \\ &= \left(1 - \frac{2\mu}{\mu + L}\right)\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \;=\; \frac{L - \mu}{L + \mu}\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2\end{aligned}$$

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 \leq \frac{\kappa - 1}{\kappa + 1}\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2$$

Linear Convergence of GD

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 &\leq \left\|\boldsymbol{I} - \tau\int_0^1 \nabla^2 f(\boldsymbol{x}(t))dt\right\|_2 \|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \\ &\leq \left[\sup_{0\leq t\leq 1}\left\|\boldsymbol{I} - \tau\nabla^2 f(\boldsymbol{x}(t))\right\|_2\right]\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2\end{aligned}$$

By the fact that f is μ-strongly convex, $\nabla^2 f(\boldsymbol{x}(t)) \geq \mu\boldsymbol{I}$, $\forall\, t$, we have

$$\begin{aligned}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 &\leq (1 - \tau\mu)\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \\ &= \left(1 - \frac{2\mu}{\mu + L}\right)\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2 \;=\; \frac{L - \mu}{L + \mu}\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2\end{aligned}$$

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \leq \langle \nabla f(\boldsymbol{x}_\star), \boldsymbol{x}_k - \boldsymbol{x}_\star\rangle + \frac{L}{2}\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2^2 \leq \frac{L}{2}\left(\frac{\kappa - 1}{\kappa + 1}\right)^{2k}\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2$$

Is SC Necessary for Linear Convergence?

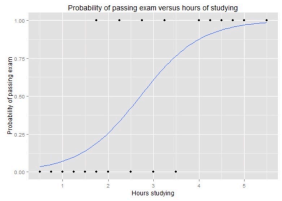
The strong convexity requirement can often be relaxed:

- Local (restricted) strong convexity;
- Regularity condition;
- Polyak-Lojasiewicz condition, etc.

Example: Logistic Regression

For binary classification, we want to learn a linear classifier to classify the data.

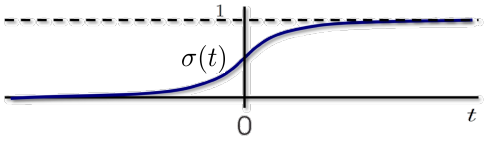
Hours z_i	Pass y_i	Hours z_i	Pass y_i
0.5	0	2.75	1
0.75	0	3.00	0
1.00	0	3.25	1
1.25	0	3.50	0
1.5	0	4.00	1
1.75	0	4.25	1
1.75	1	4.50	1
2.00	0	4.75	1
2.25	1	5.00	1
2.50	0	5.50	1



Example: Logistic Regression

- Given a bunch of training data $\{\boldsymbol{z}_i, y_i\}_{i=1}^m$ with $\boldsymbol{z} \in \mathbb{R}^n$, we want to learn the linear classifier $h(\boldsymbol{z}) = \boldsymbol{w}^\top \boldsymbol{z} + b$, so that we can make a decision by

$$y_i = \begin{cases} 1 & \text{if } \sigma(h(\boldsymbol{z}_i)) \geq 1/2, \\ 0 & \text{if } \sigma(h(\boldsymbol{z}_i)) < 1/2. \end{cases} \quad \text{with } \sigma(h(\boldsymbol{z})) = \frac{1}{1+\exp(-h(\boldsymbol{z}))}.$$



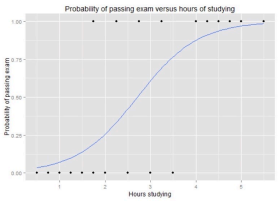
Example: Logistic Regression

Probability of passing an exam versus hours of study

Hours z_i	Pass y_i	Hours z_i	Pass y_i
0.5	0	2.75	1
0.75	0	3.00	0
1.00	0	3.25	1
1.25	0	3.50	0
1.5	0	4.00	1
1.75	0	4.25	1
1.75	1	4.50	1
2.00	0	4.75	1
2.25	1	5.00	1
2.50	0	5.50	1

$$\text{Prob.} = \frac{1}{1+\exp(-w \cdot z_i - b)}$$

$w = 1.5046$, $b = -4.0777$.



Example: Logistic Regression

For simplicity, let us assume $y_i \in \{-1, 1\}$ instead of $y_i \in \{0, 1\}$. To learn the parameter $\boldsymbol{w}$, by maximum likelihood estimation (MLE), we have

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} f(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^m \log\left(1 + \exp\left(-y_i \boldsymbol{w}^\top \boldsymbol{z}_i\right)\right)$$

$$\nabla^2 f(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^m \underbrace{\frac{\exp\left(-y_i \boldsymbol{w}^\top \boldsymbol{z}_i\right)}{\left(1 + \exp\left(-y_i \boldsymbol{w}^\top \boldsymbol{z}_i\right)\right)^2}}_{\to 0 \text{ as } \boldsymbol{w} \to \infty} \boldsymbol{z}_i \boldsymbol{z}_i^\top \overset{\boldsymbol{w} \to \infty}{\longrightarrow} \boldsymbol{0}$$

$\Longrightarrow$ f is 0-strongly convex

Does it mean that we no longer have linear convergence?

Local Strong Convexity (LSC)

Definition. (Local Strong Convexity) A function $f(\boldsymbol{x})$ is locally μ-strongly convex if it is μ-strongly convex within a ball of the global minimizer $\boldsymbol{x}_\star$: $\mathcal{B}(\delta) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{x}_\star\|_2 \leq \delta\}$

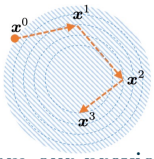
Theorem. Let f be locally μ-strongly convex and its gradient is L-Lipschitz with

$$\mu \boldsymbol{I} \preceq \nabla^2 f(\boldsymbol{x}) \preceq L \boldsymbol{I}, \quad \forall \boldsymbol{x} \in \mathcal{B}_0,$$

where $\mathcal{B}_0 := \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{x}_\star\|_2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2\}$. Then the linear convergence of GD continues to hold within $\mathcal{B}_0$.

Local Strong Convexity

$$\mathcal{B}_0 = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{x}_\star\|_2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2\}$$



- Suppose $\boldsymbol{x}_k \in \mathcal{B}_0$. From our previous analysis,
$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2 \leq \frac{\kappa - 1}{\kappa + 1} \|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2$$
- Thus, we have $\boldsymbol{x}_{k+1} \in \mathcal{B}_0$, so that we repeat the analysis for the next iteration ...

Local Strong Convexity

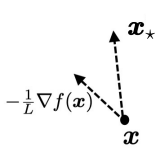
Back to the LR example, the local strong convexity parameter is controlled by

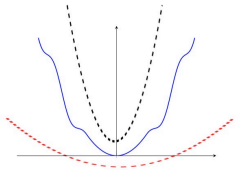
$$\mu = \inf_{\boldsymbol{w} \in \mathcal{B}_0(\boldsymbol{w}_\star)} \lambda_{\min}\left(\frac{1}{m} \sum_{i=1}^m \frac{\exp\left(-y_i \boldsymbol{w}^\top \boldsymbol{z}_i\right)}{\left(1 + \exp\left(-y_i \boldsymbol{w}^\top \boldsymbol{z}_i\right)\right)^2} \boldsymbol{z}_i \boldsymbol{z}_i^\top\right),$$

which can be strictly bounded away from 0.

For instance, when $\boldsymbol{z}_i \sim_{i.i.d.} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, we can show that $\mu \geq c_0$ for some universal constant $c_0 > 0$ if $m/n > 2$ (Sur et al.'17)

(Local) Regularity Condition



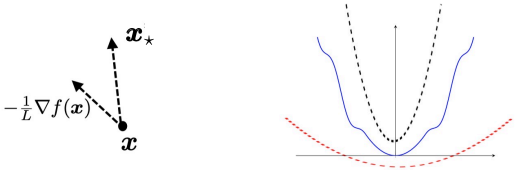


Definition. A function satisfies the regularity condition, if

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}_\star \rangle \geq \frac{\mu}{2} \|\boldsymbol{x} - \boldsymbol{x}_\star\|_2^2 + \frac{1}{2L} \|\nabla f(\boldsymbol{x})\|_2^2$$

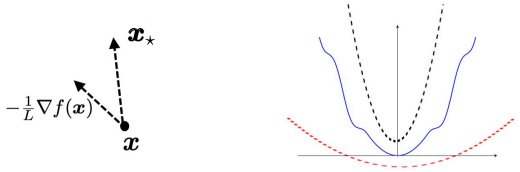
for any $\boldsymbol{x} \in \mathbb{R}^n$ or $\boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}_\star, \delta)$.

(Local) Regularity Condition



- An alternative for strong convexity and smoothness.
- Compared to strong convexity (which involves $\forall(\boldsymbol{x}, \boldsymbol{x}')$), regularity condition restrict to $(\boldsymbol{x}, \boldsymbol{x}_\star)$ for any $\boldsymbol{x}$.

(Local) Regularity Condition



Theorem. Suppose f satisfies the regularity condition. If $\tau_k = \frac{1}{L}$, then the GD method converges *linearly*

$$\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2^2 \;\leq\; \left(1 - \frac{\mu}{L}\right)^k \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2 .$$

Polyak-Lojasiewicz (PL) Condition

$$\|\nabla f(\boldsymbol{x})\|_2^2 \;\geq\; 2\mu\,(f(\boldsymbol{x}) - f(\boldsymbol{x}_\star)), \quad \forall\, \boldsymbol{x}.$$

- It guarantees that gradient $\|\nabla f(\boldsymbol{x})\|_2$ grows fast as we move away from the minimizer $\nabla f(\boldsymbol{x}_\star) = \mathbf{0}$;
- It guarantees that every stationary point is a global minimizer.

Polyak-Lojasiewicz (PL) Condition

Theorem. Suppose f satisfies the PL condition and its gradient ∇f is L-Lipschitz. Then if $\tau_k = \frac{1}{L}$, we have

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_\star) \;\leq\; \left(1 - \frac{\mu}{L}\right)^k (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_\star)) .$$

- It guarantees linear convergence to the optimal objective value;
- It does *not* imply the uniqueness of global minimizer.

Example: Overparameterized Linear Regression

Given data points $\{y_i, \boldsymbol{z}_i\}_{i=1}^m$ with $\boldsymbol{z}_i \in \mathbb{R}^n$, find a linear model that best fits the data

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} f(\boldsymbol{w}) \;=\; \frac{1}{2m}\sum_{i=1}^m \left(y_i - \boldsymbol{w}^\top \boldsymbol{z}_i\right)^2 .$$

Over-parameterization: model dimension $n >$ sample size m
— *a regime of particular interest in deep learning*

Example: Overparameterized Linear Regression

The problem is convex but *not* strongly convex, because

$$\nabla^2 f(\boldsymbol{w}) \;=\; \frac{1}{m}\sum_{i=1}^m \boldsymbol{z}_i \boldsymbol{z}_i^\top \text{ is rank-deficient if } n > m$$

But for the most "non-degenerate" cases, one has $f(\boldsymbol{w}_\star) = 0$ and the PL condition is met, and hence GD converges linearly.

Example: Overparameterized Linear Regression

Fact. Suppose that $\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{z}_1 & \cdots & \boldsymbol{z}_m\end{bmatrix}^\top \in \mathbb{R}^{m\times n}$ has rank m, and that $\tau_k = \lambda_{\max}^{-1}(\boldsymbol{Z}\boldsymbol{Z}^\top)$. Then GD obeys

$$f(\boldsymbol{w}_k) - f(\boldsymbol{w}_\star) \;\leq\; \left(1 - \frac{\lambda_{\min}(\boldsymbol{Z}\boldsymbol{Z}^\top)}{\lambda_{\max}(\boldsymbol{Z}\boldsymbol{Z}^\top)}\right)^k (f(\boldsymbol{w}_0) - f(\boldsymbol{w}_\star)), \quad \forall\, k.$$

- Very mild assumption on $\{\boldsymbol{z}_i\}_{i=1}^m$ and no assumptions on $\{y_i\}_{i=1}^m$
- There are many global minimizers, but GD has *implicit bias* to a minimizer closest to the initialization $\boldsymbol{w}_0$.

Linear Convergence without Strong Convexity

Many (local) conditions give linear rates that are weaker than strong-convexity (SC):

- 1963: Polyak-Lojasiewicz (PL)
- 1993: Error bounds
- 2000: Quadratic growth (QG)
- 2013-2015: essential SC, weak SC, restricted secant inequality, restricted SC, optimal SC, semi-SC, etc.

All of the above imply PL except QG.

Lecture Agenda

- Basics of Iterative Methods
- (Accelerated) Gradient Descent for Smooth Problems
- Linesearch Methods for Step Size
- Exploiting Function Properties for Faster Convergence
- **Newton & Quasi-Newton Method**

Newton's Direction

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}^n.$$

At the current iterate $\boldsymbol{x}_k$, consider the quadratic approximation

$$f(\boldsymbol{x}_k + \boldsymbol{d}) \approx \underbrace{f(\boldsymbol{x}_k) + \boldsymbol{d}^\top \nabla f(\boldsymbol{x}_k) + \frac{1}{2}\boldsymbol{d}^\top \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{d}}_{Q_k(\boldsymbol{d})}$$

Newton direction $\boldsymbol{d}_k^N$: minimizes $Q_k(\boldsymbol{d})$

$$\boxed{\boldsymbol{d}_k^N = -\left(\nabla^2 f(\boldsymbol{x}_k)\right)^{-1} \nabla f(\boldsymbol{x}_k).}$$

Newton's Method

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \underbrace{\tau_k}_{\text{stepsize}} \cdot \underbrace{\boldsymbol{d}_k^N}_{\text{descent direction}}.$$

- **Descent direction**: when $\nabla^2 f(\boldsymbol{x}_k) \succ \boldsymbol{0}$

$$\nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}_k^N = -\left(\boldsymbol{d}_k^N\right)^\top \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{d}_k^N < -c_k \left\|\boldsymbol{d}_k^N\right\|_2^2 < 0.$$

- **Stepsize:** natural choice is $\tau_k = 1$, but can be chosen based on Armijo linesearch

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k^N) \leq f(\boldsymbol{x}_k) + \alpha\tau \nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}_k^N.$$

Newton's Method

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \underbrace{\tau_k}_{\text{stepsize}} \cdot \underbrace{\boldsymbol{d}_k^N}_{\text{descent direction}}.$$

- **Fast Convergence:** Newton's method usually has *local quadratic convergence.*
- **Expensive:** it requires storing and inverting $\nabla^2 f(\boldsymbol{x})$.
- **Unstable:** $\nabla^2 f \succ \boldsymbol{0}$ might not hold for general nonlinear problems.

Quasi-Newton's Method

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \underbrace{\tau_k}_{\text{stepsize}} \cdot \underbrace{\boldsymbol{d}_k^{QN}}_{\text{descent direction}}.$$

$$\boldsymbol{d}_k^{QN} = -\underbrace{\boldsymbol{H}_k}_{\text{surrogate of } (\nabla^2 f(\boldsymbol{x}_k))^{-1}} \nabla f(\boldsymbol{x}_k)$$

Challenges:

- $\boldsymbol{H}_k$ needs to be a good approximation of $(\nabla^2 f(\boldsymbol{x}_k))^{-1}$
- $\boldsymbol{H}_k$ must be a descent direction with $\boldsymbol{H}_k \succ \boldsymbol{0}$
- It can be more efficiently computed than $(\nabla^2 f(\boldsymbol{x}_k))^{-1}$

Quasi-Newton's Method

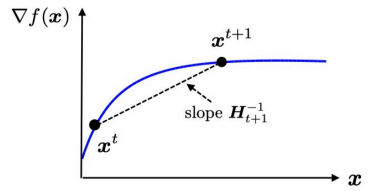
Idea: consider the following quadratic approximation of f

$$Q_H^k(\boldsymbol{x}) := f(\boldsymbol{x}_{k+1}) + (\boldsymbol{x} - \boldsymbol{x}_{k+1})^\top \nabla f(\boldsymbol{x}_{k+1}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_{k+1})^\top \boldsymbol{H}_k^{-1}(\boldsymbol{x} - \boldsymbol{x}_{k+1}).$$

Gradient matching for the last two iterates:

$$\nabla Q_H^k(\boldsymbol{x}_k) = \nabla f(\boldsymbol{x}_k), \quad \nabla Q_H^k(\boldsymbol{x}_{k+1}) = \nabla f(\boldsymbol{x}_{k+1})$$

Secant Equation



$$\boldsymbol{H}_{k+1}^{-1} \cdot \underbrace{(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)}_{\boldsymbol{s}_k} = \underbrace{\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)}_{\boldsymbol{y}_k}$$

Quasi-Newton Direction

$$\boldsymbol{H}_{k+1}^{-1} \cdot \boldsymbol{s}_k = \boldsymbol{y}_k$$

- admit an infinite number of solutions;
- $\boldsymbol{H}_k$ and $\boldsymbol{H}_{k+1}$ are close and the difference is low-rank;
- $\boldsymbol{H}_{k+1}$ needs to be positive definite

$$\boldsymbol{H}_0 \succ \boldsymbol{0}, \quad \boldsymbol{s}_k^\top \boldsymbol{y}_k > 0.$$

Quasi-Newton Direction

- **Symmetric rank-1 update (SR1):**

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{(\boldsymbol{s}_k - \boldsymbol{H}_k \boldsymbol{y}_k)(\boldsymbol{s}_k - \boldsymbol{H}_k \boldsymbol{y}_k)^\top}{(\boldsymbol{s}_k - \boldsymbol{H}_k \boldsymbol{y}_k)^\top \boldsymbol{y}_k}$$

NOT preserving positive definiteness.

- **BFGS rank-2 update:**

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{\left(\boldsymbol{s}_k^\top \boldsymbol{y}_k + \boldsymbol{y}_k^\top \boldsymbol{H}_k \boldsymbol{y}_k\right) \boldsymbol{s}_k \boldsymbol{s}_k^\top}{\left(\boldsymbol{s}_k^\top \boldsymbol{y}_k\right)^2} - \frac{\boldsymbol{H}_k \boldsymbol{y}_k \boldsymbol{s}_k^\top + \boldsymbol{s}_k \boldsymbol{y}_k^\top \boldsymbol{H}_k}{\boldsymbol{s}_k^\top \boldsymbol{y}_k}$$

BFGS Method (1970)

BFGS Method (1970)

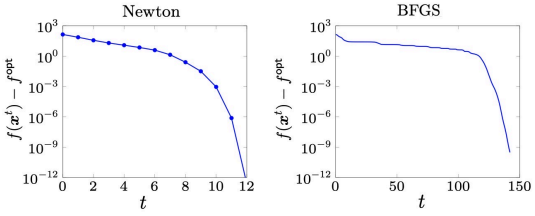
Theorem (Informal). Suppose $f \in \mathcal{C}^2$ is strongly convex, and has Lipschitz-continuous Hessian. Under mild conditions, BFGS achieves superlinear convergence

$$\lim_{k \to \infty} \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_\star\|_2}{\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2} = 0.$$

- **Iteration complexity:** from $O(n^3)$ to $O(n^2)$;
- **Asymptotic result:** holds when $k \to \infty$;
- Limited memory version reduces to $O(n)$, but only linear convergence.

BFGS Method

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



$n = 100,\ N = 500$

References & Further Readings

- *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. John Wright, Yi Ma. (**Appendix D**)
- *Numerical Optimization*, Jorge Nocedal, and Stephen Wright, Springer. (**Chapter 2, 3, & 6**)