Basic Convex Optimization2

Second-Order Method

Kyoungman Kim

Sungkyunkwan University

Department of Mathematics

MIMIC Seminar
September 23, 2025

Overview

- Convex optimization problems emerge across various disciplines. In fact, convex optimization has become one of the key tools in Machine Learning.

- There is no general analytical formula for the solution of convex optimization problmes, but there are effective methods for solving them. In this seminar, we shall restrict our discussion to Unconstrained Second-Order Method.

Definitions

Definition (Convex Function)

A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if its domain $\mathrm{dom}(f)$ is a convex set and, for all $x, y \in \mathrm{dom}(f)$ and for all $\lambda \in [0, 1]$, we have

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Remark

Geometrically, the line segment connecting

$$(x, f(x)) \quad \text{to} \quad (y, f(y))$$

should lie above the graph of the function.

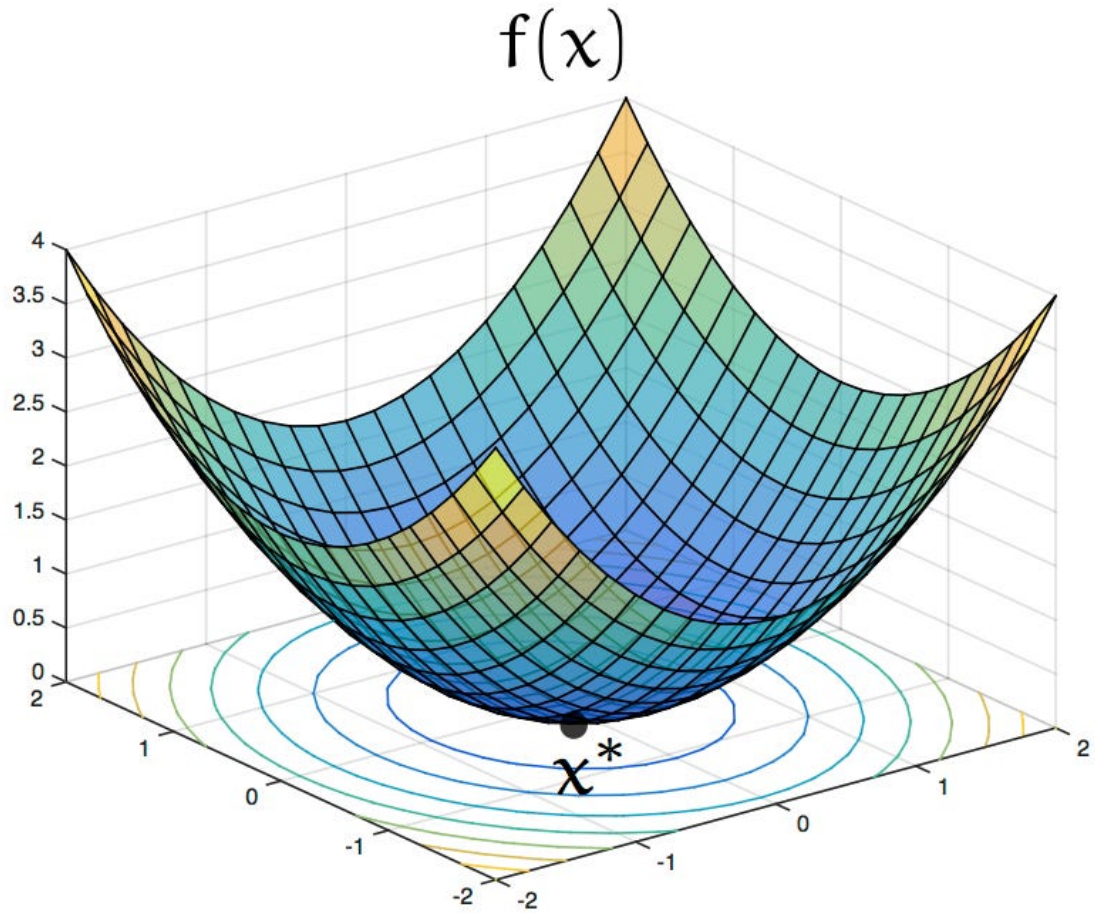


Figure: Convex Function

Nice properties : Optimality Condition

Theorem

A point $x^* \in \mathbb{R}^n$ is optimal for the unconstrained problem

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

if and only if

$$\nabla f(x^*) = 0.$$

Remark

The presence of constraints makes it significantly more challenging to establish general conditions

Example of Convex Optimization(1)

Example

A **least-squares problem** is an unconstrained optimization problem where the objective is a sum of squared terms of the form $a_i^T x - b_i$:

$$\min f_0(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{k} (a_i^T x - b_i)^2,$$

where $A \in \mathbb{R}^{k \times n}$ with $k \geq n$, each row a_i^T is a row of the matrix A, and the optimization variable is $x \in \mathbb{R}^n$.

Example of Convex Optimization (1)

Example (Analytical Solution of L-S Problem)

The optimal solution to the least-squares problem is given analytically by:

$$x^* = (A^T A)^{-1} A^T b,$$

assuming $A^T A$ is invertible (i.e., A has full column rank).

Example of Convex Optimization (2)

Example (Linear Programming (LP))

Consider the linear program:

$$\min c^T x \quad \text{subject to } a_i^T x \leq b_i, \quad i = 1, \ldots, m,$$

where $c, a_1, \ldots, a_m \in \mathbb{R}^n$ are vectors and $b_1, \ldots, b_m \in \mathbb{R}$ are scalars.

First-Order Condition for Convexity

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable over an open domain. Then the following statements are equivalent:

(i) *f is convex.*

(ii) *For all $x, y \in \mathrm{dom}(f)$,*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

- Statement (ii) is known as the *first-order condition* for convexity. It means that the tangent plane at any point lies below the graph of f.

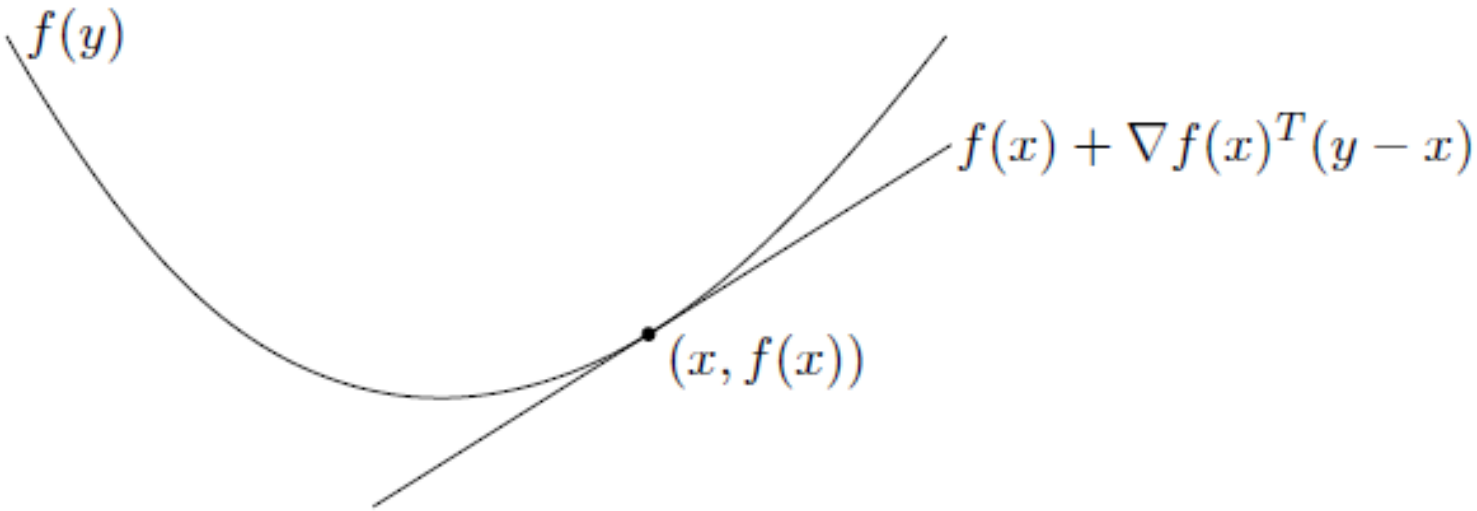


Figure: First Order Condition

Definition: Smoothness

Definition

A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be β-**smooth** if its gradient is β-Lipschitz. That is,

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\| \quad \text{for all } x, y \in \mathbb{R}^n.$$

Lemma (Smoothness Upper Bound)

Let f be a β-smooth function on $\mathbb{R}^n$. Then for any $x, y \in \mathbb{R}^n$,

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{\beta}{2} \|x - y\|^2.$$

Notation and Properties

Hessian Matrix

Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable. The Hessian is the matrix of second-order partial derivatives:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

- If $f \in C^2$ (second partials continuous), then mixed partials commute (*Clairaut*) and $\nabla^2 f(x)$ is symmetric.

Notation and Properties

Loewner Order

A **partial order** on symmetric matrices $A, B \in \mathbb{R}^{n \times n}$:

$$A \preceq B \iff B - A \succeq 0.$$

Equivalent characterizations:

- **Quadratic forms:** $v^\top A v \leq v^\top B v$ for all $v \in \mathbb{R}^n$.
- **Eigenvalues:** all eigenvalues of $B - A$ are nonnegative.

Definition: Strong Convexity and Smoothness

Definition

A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be α**-strongly convex** if for all $x, y \in \mathbb{R}^n$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2.$$

Remark

1. α-strong convexity $\iff \nabla^2 f \succeq \alpha I \iff \lambda_{\min}(\nabla^2 f) \geq \alpha$
2. β-smoothness $\iff \nabla^2 f \preceq \beta I \iff \lambda_{\max}(\nabla^2 f) \leq \beta$

Gradient Descent

Algorithm : Linear Approximation

- First-order Taylor expansion around x with quadratic regularization:

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t} \|y - x\|_2^2$$

- Minimizing over y yields the update rule:

$$x^+ = x - t\nabla f(x)$$

Remark

Each step uses a linear approximation of f with a quadratic regularization term.

Convergence of GD (Strongly Convex + Smooth)

Theorem

Let f be α-strongly convex and β-smooth on $\mathbb{R}^n$. Then gradient descent with step size

$$\eta = \frac{1}{\beta}$$

satisfies the following for all $t \geq 0$:

$$\|x_{t+1} - x^*\|^2 \leq \left(1 - \frac{\alpha}{\beta}\right)^t \|x_1 - x^*\|^2.$$

- Since the behavior appears linear on a log scale, it is referred to as linear convergence.

Second-Order Condition for Convexity

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable in an open domain. Then the following statements are equivalent:

(i) *f is convex.*

(ii) $\nabla^2 f \succeq 0$

(iii) *All eigenvalues of the Hessian are greater than or equal to zero.*

- In this seminar, we need to use the inverse of the Hessian matrix.
- Thus, we assume that $\nabla^2 f \succ 0$, which is equivalent to strict convexity.

Newton–Raphson Method for Root Finding (1)

Algorithm

Let $f : [a, b] \to \mathbb{R}$ be a differentiable function.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

This leads to

$$f\left(\lim_{n\to\infty} x_n\right) = 0$$

Newton–Raphson Method for Root Finding (2)

Algorithm

Let $f : [a, b] \to \mathbb{R}$ be a twice differentiable function.

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

This leads to

$$f'\left(\lim_{n\to\infty} x_n\right) = 0$$

- If f is convex, $f'(x) = 0$ leads to the global minimum of f.
- This idea could be generalized to the n-dimensional Newton method.

Newton Method

Algorithm : Quadratic Approximation

- Second-order Taylor expansion around x:

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x) (y - x)$$

- Minimizing over y yields:

$$x^+ = x - \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$$

Remark

Each step solves a quadratic approximation of the original problem.

Local Convergence Issue

Example

Define

$$f(x) = \sqrt{1 + x^2}.$$

Consider the problem

$$\min_{x \in \mathbb{R}} f(x).$$

Newton's iteration becomes

$$x_{t+1} = x_t - \left(f''(x_t)\right)^{-1} f'(x_t) = x_t - x_t(1 + x_t^2) = -x_t^3.$$

- No guarantee of global convergence: depends on the initial point.
- Remedy: combine Newton's method with backtracking line search.

Backtracking Line Search

Algorithm

1. Fix parameters $0 < \beta < 1$ and $0 < \alpha < \frac{1}{2}$.
2. Start with an initial step size $t = 1$.
3. While the condition

$$f(x + td) > f(x) + \alpha t \nabla f(x)^\top d$$

   is satisfied, update $t \leftarrow \beta t$, where $d = -\nabla^2 f(x)^{-1} \nabla f(x)$.
4. Use the final t as the step size.

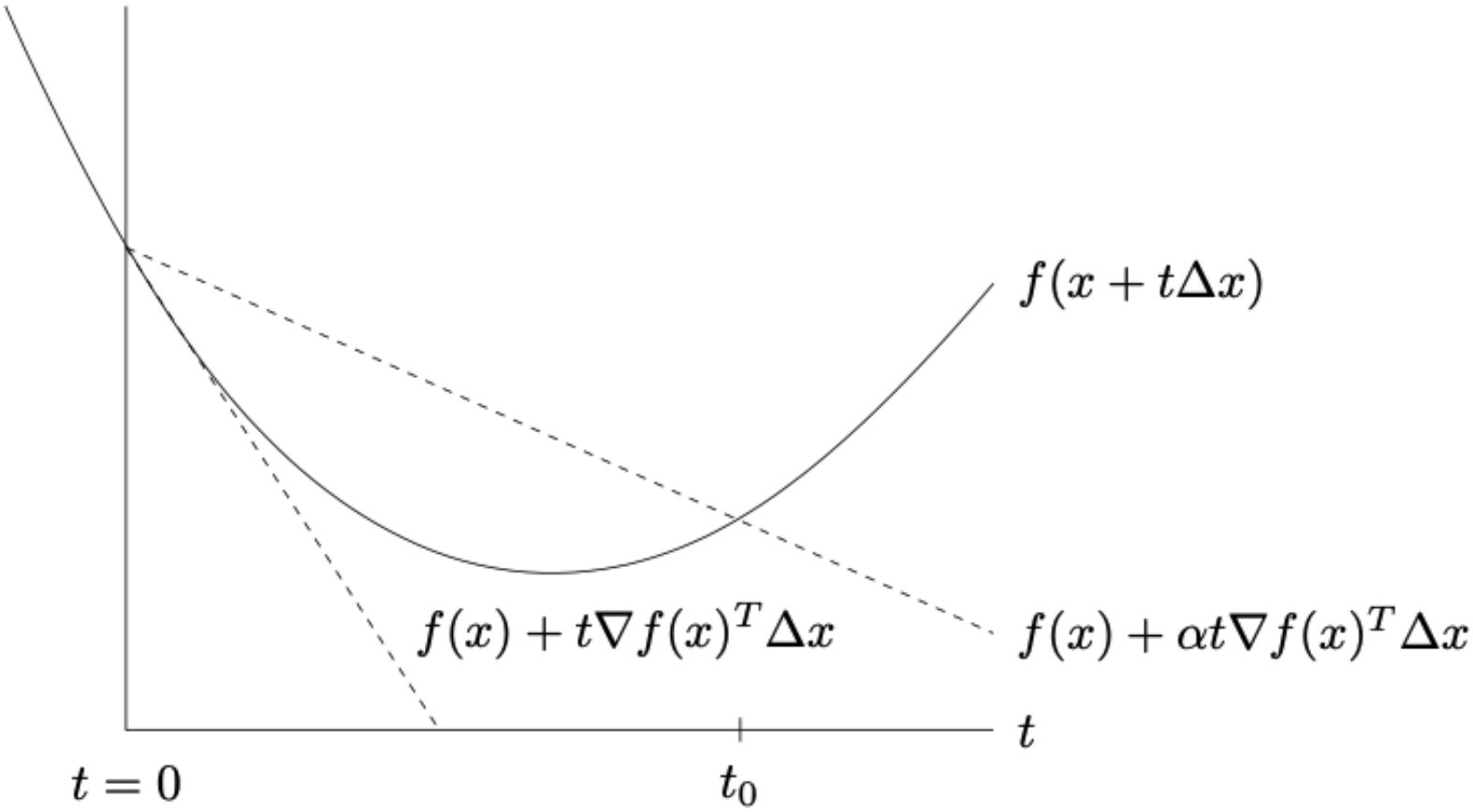


Figure: Backtracking Line Search

$$f(x + td) > f(x) + \alpha t \nabla f(x)^\top d$$

Assumptions on the Objective Function

Assumptions

Suppose the objective function f satisfies the following conditions:

- f is twice continuously differentiable.
- f is m-strongly convex in the ℓ_2 norm, i.e.,

$$\nabla^2 f(x) \succeq mI.$$

- f is M-smooth in the ℓ_2 norm, i.e.,

$$\nabla^2 f(x) \preceq MI.$$

- The Hessian of f is L-Lipschitz continuous in the ℓ_2 norm, i.e.,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2 \quad \text{(Matrix Norm)}.$$

Newton's Method(phase1)

Algorithm 1: Newton's method with backtracking line search

Initialize x_1;

for $t = 1, \ldots, T-1$ **do**

  Compute $d_t = -\nabla^2 f(x_t)^{-1} \nabla f(x_t)$;

  Compute a step size τ_t by backtracking line search;

  Update $x_{t+1} = x_t + \tau_t d_t$;

return x_T;

Newton's Method(phase2)

Algorithm 2: Pure Newton's method

Initialize x_1.;

for $t = 1, \ldots, T-1$ **do**

  Compute $d_t = -\nabla^2 f(x_t)^{-1} \nabla f(x_t)$.;

  Update $x_{t+1} = x_t + d_t$.;

return x_T.;

Convergence Analysis

Theorem

Newton's method with backtracking line search satisfies the following two-stage convergence bounds:

$$f(x_{(k)}) - f_\star \leq \begin{cases} (f(x_{(0)}) - f_\star) - \gamma k & k \leq k_0 \quad (\textit{damped phase}) \\ \dfrac{2m^3}{L^2}\left(\dfrac{1}{2}\right)^{2^{k-k_0+1}} & k > k_0 \quad (\textit{pure phase}) \end{cases}$$

Here

$$\gamma = \frac{\alpha\beta^2\eta^2 m}{L^2}, \quad \eta = \min\left\{1, \ \frac{3(1-2\alpha)m^2}{M}\right\},$$

and k_0 is the number of steps until

$$\|\nabla f(x_{(k_0+1)})\|_2 < \eta.$$

Bounding the Gradient Norm after a Newton Step

Let $\Delta x_{\text{nt}} = -[\nabla^2 f(x)]^{-1} \nabla f(x), \quad x^+ = x + \Delta x_{\text{nt}}$.

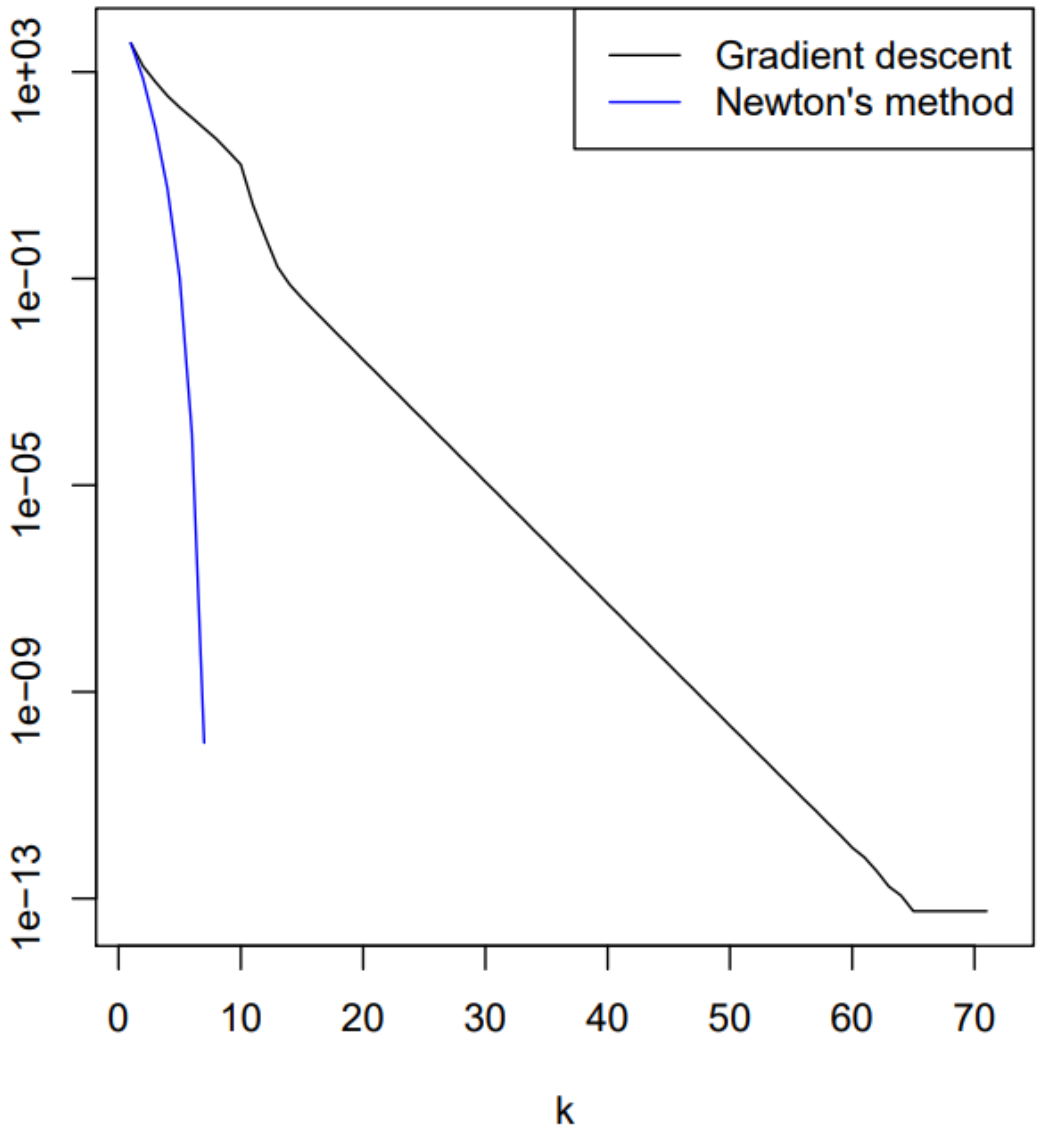
$$
\begin{aligned}
\|\nabla f(x^+)\|_2 &= \|\nabla f(x + \Delta x_{\text{nt}})\|_2 \\
&= \|\nabla f(x + \Delta x_{\text{nt}}) - \nabla f(x) - \nabla^2 f(x)\, \Delta x_{\text{nt}}\|_2 \\
&= \left\| \int_0^1 \left(\nabla^2 f(x + s\, \Delta x_{\text{nt}}) - \nabla^2 f(x)\right) \Delta x_{\text{nt}}\, ds \right\|_2 \\
&\leq \int_0^1 \left\| \nabla^2 f(x + s\, \Delta x_{\text{nt}}) - \nabla^2 f(x) \right\|_2 \|\Delta x_{\text{nt}}\|_2\, ds \\
&\leq \int_0^1 Ls \|\Delta x_{\text{nt}}\|_2^2\, ds \\
&= \frac{L}{2} \|\Delta x_{\text{nt}}\|_2^2.
\end{aligned}
$$

Quadratic Decrease of the Gradient Norm

$$\|\nabla f(x^+)\|_2 \leq \frac{L}{2}\|\Delta x_{\text{nt}}\|_2^2 = \frac{L}{2}\left\|[\nabla^2 f(x)]^{-1}\nabla f(x)\right\|_2^2 \leq \frac{L}{2m^2}\|\nabla f(x)\|_2^2$$

- Shows quadratic convergence of the gradient norm in the pure Newton phase.
- The rate depends on constant number M, m of the Hessian.
- For the complete proof, refer to *Convex Optimization* by Boyd and Vandenberghe, Chapter9.

Newton method applied to logistic regression



Why do we use the gradient to shift the phase?

Remark

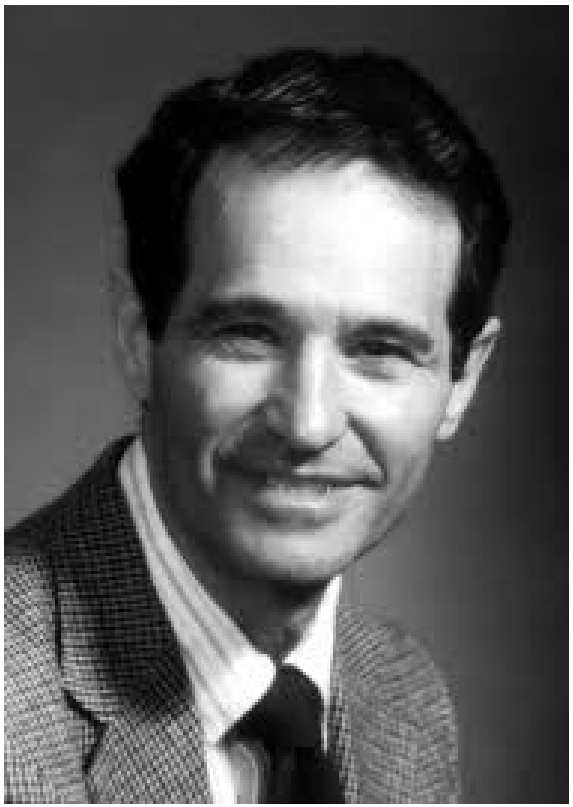
- If f is convex, $\nabla f = 0$ implies that f attains its global minimum.
- Thus, an adequately small gradient norm implies that the iterate is well prepared for the pure Newton phase.
- By using line search, we can resolve the local convergence issue. However, another serious issue remains.

Motivation for Quasi-Newton Methods

Remark

- Newton's method: high convergence rate, but costly (Hessian inverse $\mathcal{O}(n^3)$, via Cholesky decomposition).
- First-order methods: slower but simple, suitable for large-scale problems.
- Quasi-Newton: use an approximate Hessian and refine it iteratively, which reduces per-iteration cost to $\mathcal{O}(n^2)$.

The Pioneers of the Quasi-Newton Methods



W. C. Davidon

First Quasi-Newton method (1959)

R. Fletcher

Dantzig Prize (1997)

M. J. D. Powell

Dantzig Prize (1982)

Quasi-Newton (Template)

Algorithm 3: Quasi-Newton method (generic form)

Initialize $x_0 \in \mathbb{R}^n$; choose $B_0 \succ 0$;

for $k = 1, 2, 3, \ldots$ **do**

  Solve $B_{k-1}\,\Delta x_{k-1} = -\nabla f(x_{k-1})$; `// solve for` Δx_{k-1}

  Choose a step size τ_k (backtracking line search);

  Update $x_k = x_{k-1} + \tau_k\,\Delta x_{k-1}$;

  Compute B_k from B_{k-1} *(BFGS or DFP update)*;

return final iterate x_k;

How is it Quadratic?

Remark

- The 'Solve' step is actually performed in $\mathcal{O}(n^2)$ time.
- We update and store the inverse Hessian $C_{k-1} \approx B_{k-1}^{-1}$, then compute the search direction via matrix-vector multiplication:

$$\Delta x_{k-1} = -C_{k-1} \nabla f(x_{k-1})$$

- Likewise, updating the inverse Hessian from C_{k-1} to C_k (via BFGS/DFP formula) is also performed in $\mathcal{O}(n^2)$ time.

Requirements on B and B^+

Four Conditions

- **Secant equation:** $B^+ s = y$
  where $s = x^+ - x$
  and $y = \nabla f(x^+) - \nabla f(x)$.
- **Symmetry:** B^+ is symmetric.
- **Closeness:** B^+ is "close" to B.
- **Positive definiteness:** if $B \succ 0$, then $B^+ \succ 0$.

Davidon–Fletcher–Powell (DFP) Update

DFP Update

- Compute a rank-two update directly on C:

$$C^+ = C + a\,uu^\top + b\,vv^\top \quad \text{where } C = B^{-1},\ C^+ = (B^+)^{-1}.$$

- Using the secant equation $s = C^+ y$, set $u = s$, $v = Cy$, then solving for a, b gives

$$C^+ = C - \frac{C\,yy^\top C}{y^\top Cy} + \frac{ss^\top}{y^\top s}.$$

- Preserves positive definiteness; $O(n^2)$ but is less commonly used today.

Broyden–Fletcher–Goldfarb–Shanno

Figure: The four creators of BFGS (Broyden, Fletcher, Goldfarb, Shanno)

Broyden–Fletcher–Goldfarb–Shanno Update(1)

BFGS Update(1)

- Instead of a rank-two update to C, update to B

$$B^+ = B + a\,uu^\top + b\,vv^\top.$$

- Using the secant equation $B^+ s = y$ gives

$$y - Bs = (a\,u^\top s)\,u + (b\,v^\top s)\,v.$$

- Setting $u = y$, $v = Bs$ and solving for a, b we get

$$B^+ = B - \frac{Bss^\top B}{s^\top Bs} + \frac{yy^\top}{y^\top s}.$$

Broyden–Fletcher–Goldfarb–Shanno Update (2)

BFGS Update(2)

- Woodbury formula:

$$(A + UDV)^{-1} = A^{-1} - A^{-1}U\left(D^{-1} + VA^{-1}U\right)^{-1}VA^{-1}.$$

- Apply it for the inverse $C = B^{-1}$ and set

$$U = V^{\top} = \begin{bmatrix} Bs & y \end{bmatrix}, \qquad D = \begin{bmatrix} -\dfrac{1}{s^{\top}Bs} & 0 \\ 0 & \dfrac{1}{y^{\top}s} \end{bmatrix}.$$

Broyden–Fletcher–Goldfarb–Shanno Update (3)

BFGS Update (inverse form)

- After some algebra, the inverse update on C is

$$C^{+} = \left(I - \frac{sy^{\top}}{y^{\top}s}\right) C \left(I - \frac{ys^{\top}}{y^{\top}s}\right) + \frac{ss^{\top}}{y^{\top}s}.$$

- **Cost:** $O(n^2)$ per update (whereas $O(n^3)$ for exact Newton).
- **Insight:** By the Woodbury formula, the update avoids explicit matrix inversion.

Positive Definiteness of BFGS Update

Statement

- **Preservation of PD.** If $y^\top s > 0$ and $C \succ 0$, then $C^+ \succ 0$.

$$C^+ = \left(I - \frac{sy^\top}{y^\top s}\right) C \left(I - \frac{ys^\top}{y^\top s}\right) + \frac{ss^\top}{y^\top s}.$$

- **Condition** $y^\top s > 0$

$$y^\top s = \left(\nabla f(x^+) - \nabla f(x)\right)^\top (x^+ - x) > 0 \quad \text{(by strict convexity)}$$

- **Proof sketch.** For any $x \in \mathbb{R}^n$,

$$x^\top C^+ x = \left(x - \frac{s^\top x}{y^\top s} y\right)^\top C \left(x - \frac{s^\top x}{y^\top s} y\right) + \frac{(s^\top x)^2}{y^\top s}.$$

Step Length (Wolfe Conditions)

Wolfe conditions

Choose the step length t (e.g., by backtracking) so that

(i) $f(x + tp) \leq f(x) + \alpha_1 \, t \, \nabla f(x)^\top p$ (Armijo / sufficient decrease)

(ii) $\nabla f(x + tp)^\top p \geq \alpha_2 \, \nabla f(x)^\top p$ (curvature)

with $0 < \alpha_1 < \alpha_2 < 1$.

Remark

- The first condition is the same sufficient-decrease test used for Newton: it prevents t from being *too large*.
- The second condition prevents t from being *too small* and ensures adequate progress along the descent direction.

Curvature Condition

Strong Convexity Case

The curvature condition is automatically satisfied if f is strongly convex, since

$$s_k^\top y_k = \langle \nabla f(x_{k+1}) - \nabla f(x_k),\, x_{k+1} - x_k \rangle > 0,$$

General Case

For nonconvex functions, the curvature condition does not hold automatically. It is guaranteed if the step size α_k (from the previous iteration k) satisfies the second Wolfe condition.

$$\langle \nabla f(x_{k+1}),\, s_k \rangle \geq \alpha_2 \langle \nabla f(x_k),\, s_k \rangle, \qquad \alpha_2 \in (0, 1),$$

$$\langle y_k, s_k \rangle = \langle \nabla f(x_{k+1}) - \nabla f(x_k),\, s_k \rangle \geq (\alpha_2 - 1) \langle \nabla f(x_k),\, s_k \rangle > 0.$$

Superlinear Convergence

Theorem (Dennis–Moré)

Let f be twice continuously differentiable. Suppose $x_k \to x_\star$, $\nabla f(x_\star) = 0$, and $\nabla^2 f(x_\star) \succ 0$. If search directions p_k satisfy

$$\lim_{k\to\infty} \frac{\left\|\nabla f(x_k + p_k) - \nabla f(x_k) - \nabla^2 f(x_k)\, p_k\right\|}{\|p_k\|} = 0,$$

then there exists k_0 such that:

(i) the unit step $t_k = 1$ satisfies the Wolfe conditions for all $k \geq k_0$;

(ii) consequently, $x_k \to x_\star$ *superlinearly*.

Implications for Quasi-Newton Methods

DFP/BFGS under Wolfe line search

- Quasi-Newton methods generate search directions $p_k = -H_k \nabla f(x_k)$.
- Under suitable assumptions, the DFP and BFGS updates ensure the Dennis–Moré condition holds for p_k.

Notes

- For a complete proof, see *Numerical Optimization* (J. Nocedal & S. J. Wright), Chapter 8.
- Just for fun: they both received the **Dantzig Prize**, in 2012 and 2024 respectively.

Convergence of BFGS/DFP

Theorem

With line search, both BFGS and DFP converge globally. Moreover, for all $k \geq k_0$,

$$\|x_k - x_\star\|_2 \leq c_k \|x_{k-1} - x_\star\|_2,$$

where $c_k \to 0$ as $k \to \infty$.

Remark

- This is called local superlinear convergence.
- For (fixed–step) first-order methods one typically has $\|x_k - x_\star\|_2 \leq q \|x_{k-1} - x_\star\|_2$ with a *constant* q — i.e., linear convergence.

Applied to Log-Barrier Problem ($n = 100,\ m = 500$)

$$\min_{x \in \mathbb{R}^n} \quad c^\top x \ - \ \sum_{i=1}^{m} \log(b_i - a_i^\top x)$$

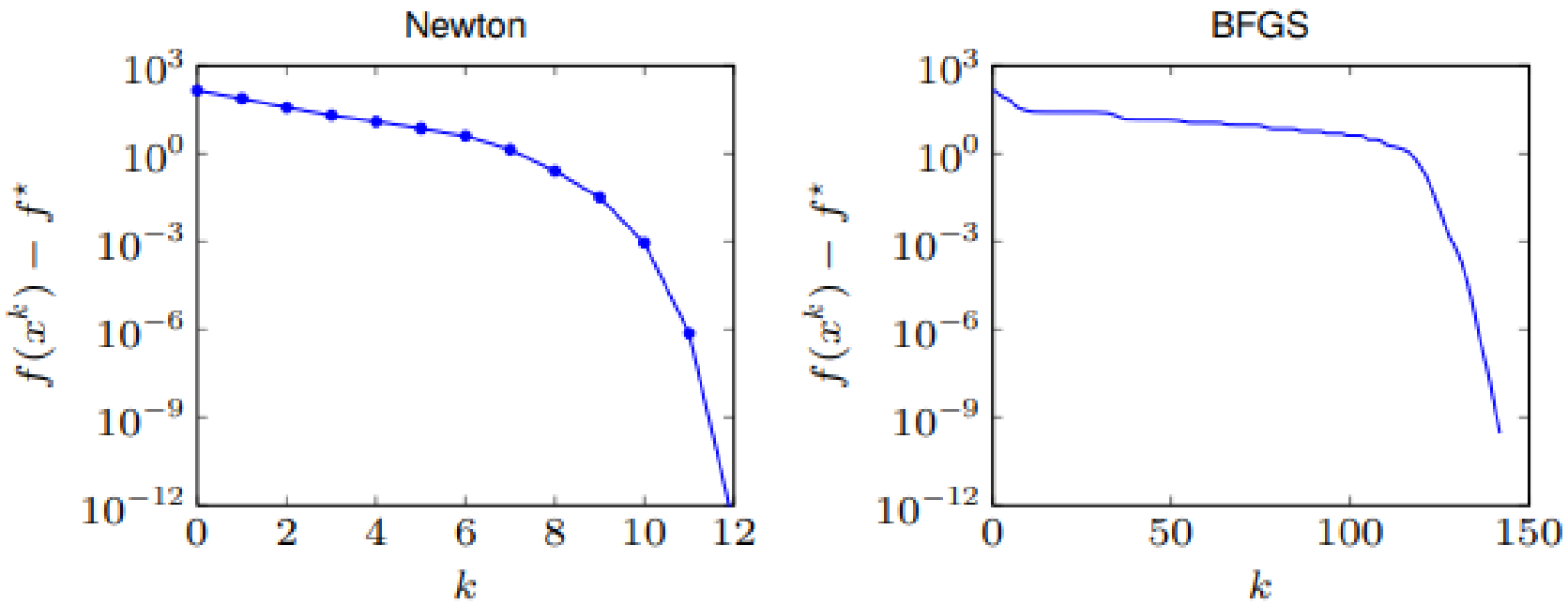


Figure: Newton vs BFGS on the log-barrier problem

Summary

Algorithm	**GD**	**NM**	**BFGS**
Convergence Rate	Linear	Quadratic	Super-linear
Cost per Iteration	$\mathcal{O}(n)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
Required Info	∇f	∇f, $\nabla^2 f$	∇f
Approximation	Linear	Quadratic	Hessian

Table: Comparison of Optimization Algorithms

The End

Questions or Comments?