Math 164: Optimization

2024-25
Fall '24

Instructor: Willem Diepeveen (+Yunuo Chen)

Textbook: E. Chong & S. Żak - An Introduction to Optimization (4th ed.)

Topics: Fundamentals of optimization, methods for unconstrained optimization, solutions to linear systems, linear programming & duality for LPs, nonlinear programming

Table of Contents

(0) Review - 100

(1) Basics of Optimization
- (i) 1st-Order Conditions - 104
- (ii) 2nd-Order Conditions - 105

(2) 1D Line Search
- (i) 1D Line Search - 106
- (ii) Newton's Method - 108

(3) Gradient Methods
- (i) Gradient & Steepest Descent - 111
- (ii) Convergence of Gradient Methods - 114
- (iii) Newton's Method for $\mathbb{R}^n$ - 119
- (iv) Convergence of Newton's Method - 121

(4) Conjugate Direction Methods
- (i) Conjugate Direction Methods - 124
- (ii) Quasi-Newton Methods - 125

(5) Solving Linear Systems
- (i) Recursive Least Squares - 128
- (ii) Minimum-Norm Solutions - 130
- (iii) The Moore-Penrose Inverse - 131

(6) Linear Programming
- (i) Intro to Linear Programming - 132
- (ii) Basic Solutions to LPs - 133
- (iii) The Simplex Algorithm - 135
- (iv) Duality - 138

(7) Intro to Nonlinear Programming
- (i) Lagrange's Theorem - 141
- (ii) The KKT Conditions - 144
- (iii) Algorithms for Optimization - 147

Linear Algebra Review

9/27/24

Lecture 1 + Disc. 1

Linear Algebra Recap

Def: A linear transformation is a function $L: \mathbb{R}^n \to \mathbb{R}^m$ that satisfies $L(ax+y) = aL(x) + L(y)$ $\forall x, y \in \mathbb{R}^n$, $a \in \mathbb{R}$; for any basis, $\exists$ a matrix $A \in \mathbb{R}^{m \times n}$ s.t. $L(x) = Ax$ $\forall x \in \mathbb{R}^n$.

- For any $A \in \mathbb{R}^{m \times n}$, call $\lambda \in \mathbb{R}$ an eigenvalue of A with (nonzero) eigenvector $v \in \mathbb{R}^n$ if $Av = \lambda v$; equivalently, $\det(A - \lambda I) = 0$. [$\to \det(A) = \prod_{i=1}^{n} \lambda_i$; $\operatorname{trace}(A) = \sum_{i=1}^{n} \lambda_i$]

  → ○ If A has n (linearly) independent eigenvectors $v_1, \dots, v_n$, can write $A = T \Lambda T^{-1}$, where $T = (v_1 \cdots v_n)$ and $\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$.

  → ○ If A is symmetric, then all eigenvalues will be real.

Orthogonal Projections

Say that two vectors $x, y \in \mathbb{R}^n$ are orthogonal if $x \cdot y = \langle x, y \rangle = 0$.

→ **Def:** For any linear subspace $V \subseteq \mathbb{R}^n$, define its orthogonal complement $V^\perp$ as $V^\perp = \{x \in \mathbb{R}^n : \langle x, y \rangle = 0 \ \forall y \in V\}$. [Observe: $\mathbb{R}^n = V \oplus V^\perp$]

- Call a linear transformation B an orthogonal projection onto a subspace V if $\forall x \in \mathbb{R}^n$, $Bx \in V$ and $x - Bx \in V^\perp$.

(*) Matrix Norms

A matrix norm is a function $\|\cdot\| : \mathbb{R}^{m \times n} \to \mathbb{R}$ satisfying [$\forall A, B \in \mathbb{R}^{m \times n}$]:

(i) $\|A\| > 0$ if $A \neq 0$, and ≥ 0 always
(ii) $\|cA\| = |c| \cdot \|A\|$ $\forall c \in \mathbb{R}$
(iii) $\|A + B\| \leq \|A\| + \|B\|$
} General norm axioms

(iv) $\|AB\| \leq \|A\| \cdot \|B\|$ — matrix norm-specific

(*) **Ex:** $\|A\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$ (Fubini's norm)

Quadratic Forms

9/27/24

Quadratic Forms

Lecture 1 + Disc. 1 + Lecture 2

Def: A quadratic form is a function $f: \mathbb{R}^n \to \mathbb{R}$ satisfying $f(x) = x^T A x \ \forall\ x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{n \times n}$. (By convention: A is symmetric.)

Definiteness of x^TAx	Value of x^TAx	Eigenvalues λ_i of A
Positive definite / PD	$f(x) > 0 \ \forall\ x \neq 0$	$\lambda_i > 0 \ \forall\ i$
Positive semi-def. / PSD	$f(x) \geq 0 \ \forall\ x$	$\lambda_i \geq 0 \ \forall\ i$
Negative definite / ND	$f(x) < 0 \ \forall\ x \neq 0$	$\lambda_i < 0 \ \forall\ i$
Negative semi-def. / NSD	$f(x) \leq 0 \ \forall\ x$	$\lambda_i \leq 0 \ \forall\ i$
Indefinite	$\exists\ x_1, x_2$ s.t. $\begin{cases} f(x_1) > 0 \\ f(x_2) < 0 \end{cases}$	$\exists\ \lambda_i > 0, \lambda_i < 0$

(*) Sylvester's Criterion

Let $A = (a_{ij})_{n \times n}$ symmetric $\to$ define its leading principal minors $d_1 = a_{11}$, $d_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$, ..., $d_n = \det(A)$ [d_k = determinant of upper-left $k \times k$ submatrix].

$\to$ • A is positive-definite $\iff d_k > 0 \ \forall\ k = 1, \dots, n$

$\to$ • A is negative-definite $\iff (-1)^k d_k > 0 \ \forall\ k = 1, \dots, n$ [$d_1 < 0, d_2 > 0, d_3 < 0, \dots$]

(*) No such statement for PSD / NSD forms

Calculus Review

Let $f: \Omega \subseteq \mathbb{R}^n \to \mathbb{R}^m \implies$ denote the differential of f at a point $x \in \mathbb{R}^n$ as $D_x f: \mathbb{R}^n \to \mathbb{R}^m$.

• $D_x f$ is a linear operator $\to$ has a matrix representation (Jacobian):

$$J_f = \left[\frac{\partial f}{\partial x_1} \ \dots \ \frac{\partial f}{\partial x_n} \right]^T = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{m \times n} \longrightarrow D_x f = J_f x$$

Calculus Recap

9/29/24

Lecture 2 + Disc. 2

Calculus Review (cont.)

Let $f: \mathbb{R}^n \to \mathbb{R} \Rightarrow$ define:

(i) Gradient of f: $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix}^T \in \mathbb{R}^n \qquad (\nabla f(x) = D_x f^T)$

(*) Directional derivative of $x \in \mathbb{R}^n$: $= \langle d, \nabla f(x) \rangle$ [$\|d\| = 1$]

(ii) Hessian of f:

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

(*) $\nabla^2 f(x)$ PD everywhere $\Rightarrow f$ is convex

Chain Rule: $f: \mathbb{R}^n \to \mathbb{R}^m$, $g: \mathbb{R}^m \to \mathbb{R}^p \Rightarrow D_x(g \circ f) = D_{f(x)} g \circ D_x f$

$= (D_x f)^T \nabla g(f(x))$ [$p = 1$]

Product Rule: $f, g: \mathbb{R}^n \to \mathbb{R}^m$; $h: \mathbb{R}^n \to \mathbb{R}$, $h(x) = f(x) \cdot g(x)$

$\Rightarrow D_x h = f(x)^T D_x g + g(x)^T D_x f$

Taylor Expansion

Let $f: \mathbb{R}^n \to \mathbb{R}$ differentiable [$f(x) \in \mathbb{R}$, $\nabla f(x) \in \mathbb{R}^n$, $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$], $x \in \mathbb{R}^n$:

(last term: want to minimize)

$$\Rightarrow \forall\, y \in \mathbb{R}^n: \quad \boxed{f(y) = f(x) + \nabla f(x)^T (y - x) + (y - x)^T \nabla^2 f(x)(y - x) + O(\|y - x\|^3)}$$

(*) Level Sets

Let $f: \mathbb{R}^n \to \mathbb{R} \Rightarrow$ define level set of f at height $c \in \mathbb{R}$ as $L_c = \{x \in \mathbb{R}^n : f(x) = c\}$

- The gradient ∇f is always orthogonal to any level sets of f [$\forall\, x \in L_c$, $x \cdot \nabla f(x) = 0$]

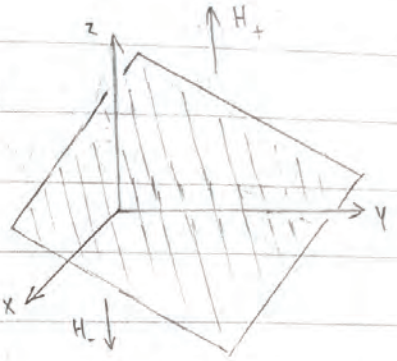
Geometry & Topology Review

9/24/24
Lecture 2 (cont.)

Geometry Recap

Def: A hyperplane is a set $H = \{x \in \mathbb{R}^n : \langle u, x \rangle_2 = v\}$ for some $u \in \mathbb{R}^n$, $v \in \mathbb{R}$ [$\dim(H) = n-1$]

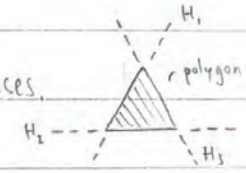
- Define its positive half-space as $H_+ = \{x \in \mathbb{R}^n : \langle u, x \rangle \geq v\}$ and negative half-space as $H_- = \{x \in \mathbb{R}^n : \langle u, x \rangle \leq v\}$



Def: A linear variety is a set $\{x \in \mathbb{R}^n : Ax = b\}$ for some $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

Def: A polytope is a set that can be expressed as an intersection of half-spaces.

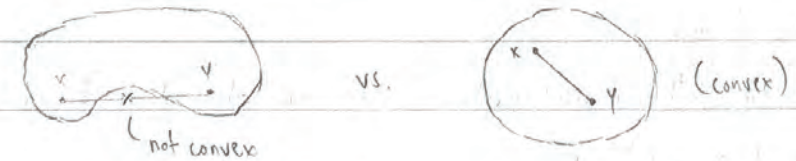
- A polygon is a bounded polytope.



Topology Recap

Def: A set S is called convex if $\forall x, y \in S$: $\alpha x + (1-\alpha) y \in S \quad \forall \alpha \in [0, 1]$

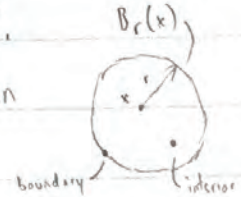
(*) Ex: Line segments, linear subspaces, half spaces, intersections of convex sets

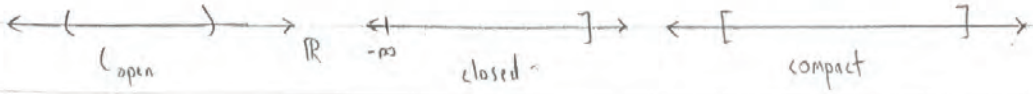


Def: A neighborhood of $x \in \mathbb{R}^n$ is a set $B_\varepsilon(x) = \{y \in \mathbb{R}^n : \|y - x\| \leq \varepsilon\}$ for some $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$.

Def: Given a set S, a point $x \in S$ is called an interior point if $\exists\, \varepsilon > 0$ s.t. $B_\varepsilon(x) \subseteq S$; otherwise, x is called a boundary point if $\forall\, \varepsilon > 0$, $B_\varepsilon(x) \cap S \neq \emptyset$ and $B_\varepsilon(x) \cap S^c \neq \emptyset$.

- S is called "open" if all of its points are interior; "closed" if $S^c = \mathbb{R} \setminus S$ is open
- S is called bounded if $S < B_R(\vec{0})$ for some $R \in \mathbb{R}$
- S is called compact if S is closed & bounded
  - S compact $\Rightarrow$ any continuous function will achieve their suprema (max/min) on S





Basics of Optimization

10/2/24

Lecture 3

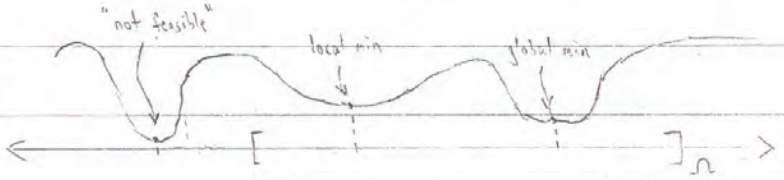
Intro to Optimization

Want to solve problems of the form: "find $\min f(x)$ subject to $x \in \Omega$", for some objective function $f: \mathbb{R}^n \to \mathbb{R}$ and constraint set $\Omega \subseteq \mathbb{R}^n$.

- Not immediately clear whether a minimizer exists or is unique

Def: A point $x^* \in \Omega$ is called a local minimizer of f under Ω if $\exists \varepsilon > 0$ s.t. $f(x) \geq f(x^*)$ $\forall\, x \in \Omega \cap B_\varepsilon(x^*)$; a strict local minimizer if $f(x) > f(x^*)$ $\forall\, x \in (\Omega \cap B_\varepsilon(x^*)) \setminus \{x^*\}$.

Def: A point $x^* \in \Omega$ is called a global minimizer if $f(x) \geq f(x^*)$ $\forall\, x \in \Omega$; a strict global minimizer if $f(x) > f(x^*)$ $\forall\, x \in \Omega \setminus \{x^*\}$.



Necessary Conditions for Local Minimizers

Def: Let $\Omega \subseteq \mathbb{R}^n$. A vector $d \in \mathbb{R}^n$ ($\neq 0$) is called a feasible direction of $x \in \Omega$ if $\exists\, \alpha_0 > 0$ s.t. $x + \alpha d \in \Omega$ $\forall\, \alpha \in [0, \alpha_0]$.

Thm (First-Order Necessary Conditions)

Let $\Omega \subseteq \mathbb{R}^n$, $f \in C^1(\Omega)$. If $x^* \in \Omega$ is a local minimizer of f, then for any feasible direction d of x^*, have that:

$$\boxed{\langle d, \nabla f(x^*) \rangle \geq 0}$$

Corollary: If x^* is a local minimizer and an interior point, then:

$$\boxed{\nabla f(x^*) = 0}$$

Basics of Optimization (cont.)

10/2/24
Lecture 3 + Lecture 4

Conditions for Local Minimizers (cont.)

(*) Proof of 1st-Order Necessary Conditions:

Let $x \in \Omega$ be a local minimizer, $d \in \mathbb{R}^n$ feasible direction of x^* with $\alpha_0 > 0$. Define $x: [0, \alpha_0] \to \mathbb{R}^n$ by $x(\alpha) = x^* + \alpha d$ and consider $\phi = f \circ x$ $(\phi: [0, \alpha] \to \mathbb{R})$.

→ By Taylor's theorem:

$$0 \leq f(x^* + \alpha d) - f(x^*) = \phi(\alpha) - \phi(0)$$
$$= \phi'(0)\alpha + o(\alpha)$$
$$= \alpha \cdot d^T \nabla f(x^*) + o(\alpha) \quad \text{↝ pick } \alpha_0 \text{ small.}$$ ▨

Thm. (Second-Order Necessary Conditions)

Let $f \in C^2(\Omega)$. Let x^* be a local min and $d \in \mathbb{R}^n$ feasible direction of x^*.

If $d^T \nabla f(x^*) = 0$ [is vanishing], then:

$$\boxed{d^T D_x^2 f(x^*) d \geq 0}$$

2 (*) Proof: Via Taylor's thm.

Corollary: If x^* is interior, then $\nabla f(x^*) = 0$ and $d^T D_x^2 f(x^*) d \geq 0 \ \forall \ d \in \mathbb{R}^n$.

(Informally: f behaves locally as a positive-definite quadratic form [paraboloid].)

Thm. (Second-Order Sufficient Conditions)

Let $f \in C^2(\Omega)$, defined on a region on which $x^* \in \Omega$ is an interior point. If:

(1) $\nabla f(x^*) = 0$, and

(2) $D^2 f(x^*) > 0$ $[\Leftrightarrow d^T D^2 f(x^*) d > 0 \ \forall \ d \in \mathbb{R}^n]$

⇒ Then x^* is a <u>strict</u> local minimizer of f.

(*) Proof: Via Rayleigh's inequality & Taylor's thm

1D Line Search

10/4/24

Lecture 4 (cont.)

One-Dimensional Line Search

Want to minimize functions $f: \mathbb{R} \to \mathbb{R}$ (precursor to multivariate case)

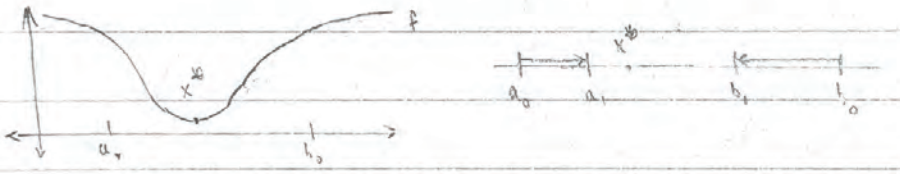
- Different algorithms for different types of information: (i) only f, [0th order] (ii) only f', [1st order] (iii) f' and f'' [2nd order]

(i) Golden Section [0th order]

(closed & bounded $\Rightarrow$ compact)

Goal: Want to minimize $f: \mathbb{R} \to \mathbb{R}$ over a closed interval $[a_0, b_0]$

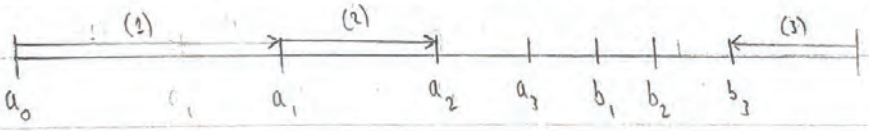
- Ideally: want to make as few evaluations $f(x)$ as possible to find approximate minimizer
- Assume f is unimodal - only has one local minimizer (in the domain)



$\rightsquigarrow$ Approach: Keep picking $a_1, b_1 \in \mathbb{R}$, $a_0 \le a_1 < b_1 \le b_0$ in order to shrink the size of the search space around the min. point x^*.

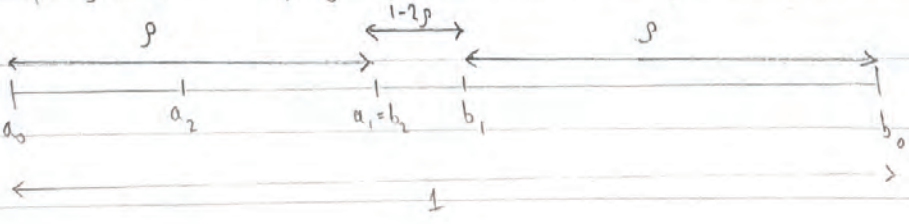
Naive Algorithm: For $\rho < 1/2$, pick $a_1, b_1 \in \mathbb{R}$ s.t. $a_1 - a_0 = b_1 - b_0 = \rho(b_0 - a_0)$. Take whichever of a_1/b_1 has greater f-value as the new left/right endpoint & repeat.

(*) Ex:



Notice: Can try to "reuse" the b_1, b_2, a_3, etc. measurements that didn't result in a "shift"

$\rightarrow$ Can pick ρ s.t. $b_1 = a_2$, e.g. $[\rho(a_1 - b_0) = a_1 - a_2]$; only 1 f-evaluation/iteration



1D Line Search (cont.)

10/4/24

Lecture 4 + Lecture 5

(i) Golden Section (cont.)

"Golden section": $\frac{\rho}{1-\rho} = \frac{1-\rho}{1}$

To satisfy $b_1 = a_2$, need that: (1) $a_1 - b_0 = 1-\rho$, (2) $a_1 - a_2 = 1-2\rho$

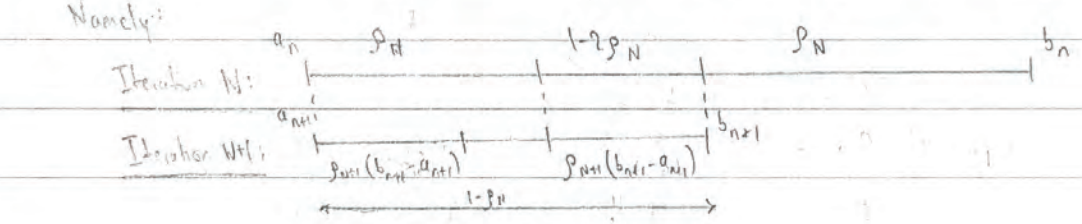
$\Rightarrow \rho(1-\rho) = 1-2\rho$; solutions: $\rho = \frac{3 \pm \sqrt{5}}{2}$ $\xrightarrow{\rho < 1/2}$ $\boxed{\rho^* = \frac{3-\sqrt{5}}{2}}$

(ii) Performance: Size of box after N iterations is $(1-\rho)^N \approx 0.61803^N$

(i) Fibonacci Method [0th Order]

Idea: Rather than fixed ρ (Golden Section), can have a different ρ for each iteration.

Namely:



$\Rightarrow \rho_{N+1}(1-\rho_N) = 1-2\rho_N \iff \rho_{N+1} = 1 - \frac{\rho_N}{1-\rho_N}$ ← Want to find $\rho_1, \rho_2, \ldots$ satisfying this

$\rho_1 = \rho_2 = \ldots \rho_N = \frac{3-\sqrt{5}}{2}$ is a solution; want the "best" solution (minimizing approximation error after N iterations):

→ Problem:

$$\begin{aligned} &\text{minimize} && (1-\rho_1)(1-\rho_2)\ldots(1-\rho_N) \\ &\text{subject to} && \rho_{n+1} = 1 - \frac{\rho_n}{1-\rho_n}, && n = 1, \ldots, N-1 \\ &&& 0 \le \rho_n \le 1/2, && n = 1, \ldots, N \end{aligned}$$

→ A solution: $F_1, F_2, \ldots, F_N \in \mathbb{N}$ Fibonacci numbers $[F_{n+1} = F_n + F_{n-1}\ ;\ F_0 = F_1 = 1]$

$$\rightarrow \rho_1 = 1 - \frac{F_N}{F_{N+1}},\ \rho_2 = 1 - \frac{F_N}{F_{N-1}},\ \ldots,\ \rho_N = 1 - \frac{F_1}{F_2}$$

Error: $(1-\rho_1)\cdot\ldots\cdot(1-\rho_N) = \frac{F_N}{F_{N+1}} \cdot \frac{F_{N-1}}{F_N} \cdot \ldots \cdot \frac{F_1}{F_2} = F_1 / F_{N+1} = \underline{1/F_{N+1}}$.

Newton's Method

10/7/24
Lecture 5 + Lecture 6

(ii) Bisection Method [1st Order]

Can search much faster by using gradient information (f')

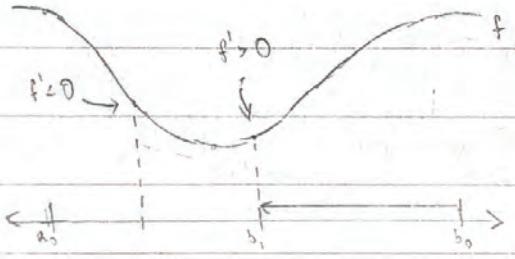
→ Idea: Perform binary search for $f' = 0$ [assuming f is unimodal]

Alg.: At every iteration, evaluate $f'\left(\frac{b_n + a_n}{2}\right)$

- If $f' > 0$, min. is to the left → $a_{n+1} = a_n$, $b_{n+1} = \frac{b_n + a_n}{2}$
- If $f' < 0$, min. is to the right → $a_{n+1} = \frac{b_n + a_n}{2}$, $b_{n+1} = b_n$

} Error: $\frac{1}{2^N}$

→ Advantages: Faster convergence than GS/Fibonacci ($\frac{1}{2^N}$ vs. $(1-\rho^{\#})^N$, $\frac{1}{F_{n+1}}$), simpler to perform



(iii) Newton's Method [2nd Order]

Idea: Since the function is assumed "simple" (e.g. unimodal), can we make it even simpler?

→ Approach: Can approximate f up to some order using a Taylor series

- In particular, can use a 2nd-order approximation to guarantee a solution [provided $f''(x) > 0$]

Approximate f by $q: \mathbb{R} \to \mathbb{R}$ defined by:

$$q(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2$$

→ q has minimum at x where $q'(x) = 0$ [FONC]:

$$0 = q'(x) = f'(x_0) + f''(x_0)(x - x_0) \iff \boxed{x = x_0 - \frac{f'(x_0)}{f''(x_0)}}$$

[(approximate soln.)]

Newton's Method (cont.)

10/9/24

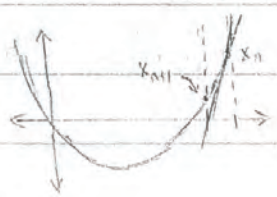
Lecture 6 (cont.)

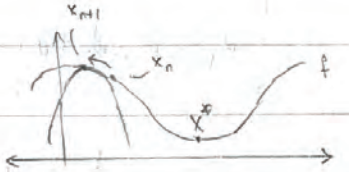
(iii) Newton's Method (cont.)

Newton's method: Iteratively take approximations $\boxed{x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}}$

(*) Remarks

(1) Newton's method works well when $f''(x) > 0$ everywhere; may fail if $f''(x) < 0$ for some $x \in \mathbb{R}$





(2) Newton's method more generally used as a method for finding zeros of a function (in our case: $g = f'$); also referred to as Newton's method for tangents

(ii) Secant Method [1st Order]

If f'' is not available (for Newton's method), can approximate f'' by $\boxed{f''(x^k) \approx \frac{f'(x^k) - f'(x^{k-1})}{x^k - x^{k-1}}}$

→ Secant method: Iteratively take approximations $\boxed{x^{k+1} = x^k - \frac{x^k - x^{k-1}}{f'(x^k) - f'(x^{k-1})} f'(x^k)}$

(ii)

(*) Note: This is analogous to Newton's method, but utilizing secants rather than tangents

In practice: both Newton's method & secant method converge faster than bisection, though neither is guaranteed to work [find the minimum]

- Can use golden section/Fibonacci/bisection search to find a region where $f''(x) > 0$, then use Newton's method/secant method inside

(*) 1D Line Search

10/9/24
Lecture 6 (cont.)

(*) Line Search in Multiple Dimensions

1D line search algorithms appear in higher-dimensional searches as well - typically, used for finding a step size $\alpha^k \geq 0$ at every iteration:

$$\alpha_k = \min_{\alpha > 0} \phi_k(\alpha) = f(x^k + \alpha d^k) \longrightarrow x^{k+1} = x^k + \alpha_k d^k \in \mathbb{R}^n$$

(d = search direction)

In most cases: don't need (or want) an exact min [for α], only an approximate one.

- Performing more iterations to update x^* more helpful than finding exact values for α^k

→ Want to make sure α^k neither too small nor too large:

(1) Armijo condition: Let $\varepsilon \in (0,1)$, $\gamma > 1$, $\eta \in (\varepsilon, 1)$:

(i) $\phi_k(\alpha_k) \leq \phi_k(0) + \varepsilon \alpha_k \phi_k'(0)$ [not too large]

(ii) $\phi_k(\gamma \alpha_k) \geq \phi_k(0) + \varepsilon \gamma \alpha_k \phi_k'(0)$ [not too small]

(2) Goldstein condition:

(i) Same as Armijo condition

(ii) $\phi_k(\alpha_k) \geq \phi_k(0) + \eta \alpha_k \phi_k'(0)$

Simple algorithm fulfilling conditions - backtracking

- Start with some α^0
- Let $\tau \in (0,1)$ (dep on choice of condition)
- Use $\alpha^k = \alpha^0 \tau^m$, where m is the smallest integer s.t. α^k satisfies the conditions

(*) Newton's Method for Tangents

For finding roots (not necessarily minima) of a function f, use Newton's method

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

(*) Can find analogous form for secant method

→ "Newton's method for optimization": Newton's method, used to find roots of $g = f'$ [not f]

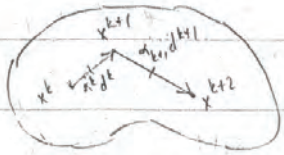
Start-Up Gradient Methods

10/11/24

Lecture 7 + Disc. 3

Multi-Dimensional Optimization

Goal: Given $f: \mathbb{R}^n \to \mathbb{R}$, want to minimize f without constraints (i.e. $\Omega = \mathbb{R}^n$)



→ Use an iterative method: $\vec{x}^{(0)}, \vec{x}^{(1)}, \dots, \vec{x}^{(k)} \xrightarrow{(?)} \vec{x}^*$

$$\vec{x}^{(k+1)} = \vec{x}^k + \alpha^{(k)} \vec{d}^{(k)} \quad [\text{Goal: } \vec{x}^{(k)} \to \vec{x}^{(*)}]$$

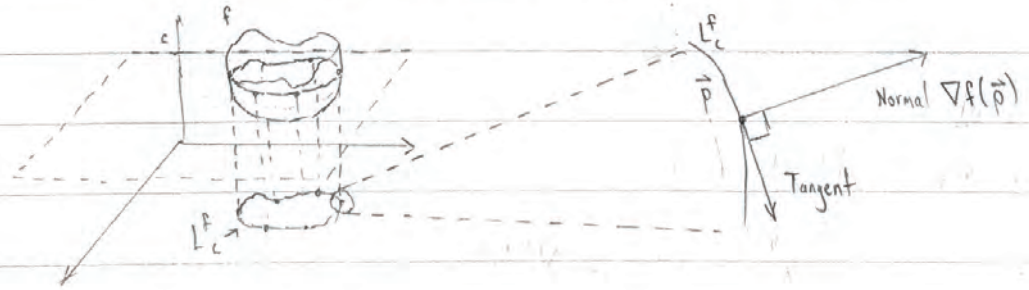
$\alpha^{(k)}$: Step size; $\vec{d}^{(k)}$: Search direction

⇒ **Q**: How to choose the search direction $\vec{d}^{(k)}$?

- **Approach**: Can try to find a descent direction: a vector $\vec{d} \in \mathbb{R}^n$ s.t. $\exists \alpha_0 > 0$ s.t. $f(\vec{x}^{(k)} + \alpha \vec{d}^{(k)}) < f(\vec{x}^{(k)}) \quad \forall \alpha \in (0, \alpha_0)$

Start-Up Gradient Methods

Recall: A level set is a set $L_c^f = \{x \in \mathbb{R}^n : f(x) = c\}$ [function f, height c]



Fact: For any $x \in L_c^f$, $\nabla f(x)$ is a normal vector to L_c^f at x

→ **Prop**: Let $f \in C^1$; then for any $d \in \mathbb{R}^n$, $d^T \nabla f(\vec{x}) < 0 \Rightarrow \vec{d}$ is a descent direction at $\vec{x}$.

Approach: Use the negative gradient $-\nabla f(\vec{x})$ as the search direction from a point $\vec{x} \in \mathbb{R}^n$.

Gradient Descent

10/11/24

Lecture 7 (cont.)

(*) Start-Up Gradient Methods (cont.)

Claim: $-\nabla f(x)$ is a descent direction at x

Proof: Let $d^* = -\nabla f(x)$; want to show that for "proper" $\alpha_0 > 0$, $f(x - \alpha_0 \nabla f(x)) < f(x)$

→ By Taylor theorem:

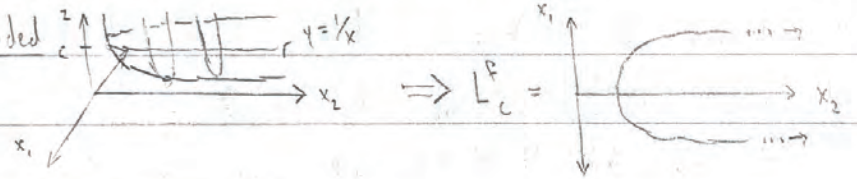
$$f(x - \alpha_0 \nabla f(x)) = f(x) - \alpha_0 \nabla f(x)^T \nabla f(x) + \mathcal{O}(\alpha_0)$$

$$= f(x) - \alpha_0 \underbrace{\|\nabla f(x)\|^2}_{>0} + \mathcal{O}(\alpha_0)$$

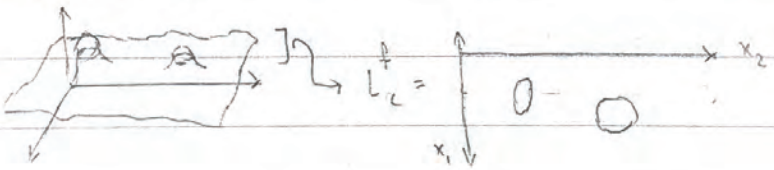
$< f(x)$ for α_0 small enough. ▨

(*) In general, may need to consider level sets that are (e.g.)

(i) Unbounded



(ii) Disconnected:



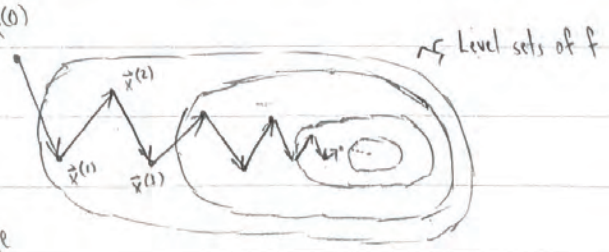
Gradient Descent

For "proper" $\alpha_k \geq 0$ $[k = 1, 2, \ldots]$:

$$\boxed{x^{k+1} = x^k - \alpha_k \nabla f(x^k)}$$

→ **Q:** What is a "proper" α_0?

- Too small → inefficient/slow
- Too large → zigzagging, may not converge



Steepest Descent

10/11/24

Lecture 7 (cont)

Steepest Descent

For each k, want to find α_k that achieves the max possible descent

→ Approach (Steepest descent): Perform a 1D line search for α_k:

$$\boxed{\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} f\left(x^k - \alpha \nabla f(x^k)\right)}$$

($k-1)^{th}$ jump only ends where it is parallel to $L^f_{f(x^k)}$; k^{th} jump is perpendicular to $L^f_{f(x^k)}$

(*) Observations: (i) For each k, $x^{k+1} - x^k \perp x^k - x^{k-1}$

(ii) As long as $\nabla f(x^k) \neq \vec{0}$, $f(x^{k+1}) < f(x)$

Stopping Criteria

Ideally, want to stop where $\nabla f(x^k) = 0$ → in practice, rarely happens

Instead, can utilize various stopping criteria:

(1) $\|\nabla f(x^k)\| < \varepsilon$ [For some choice of $\varepsilon > 0$]

(2) $|f(x^{k+1}) - f(x^k)| < \varepsilon$

(3) $\|x^{k+1} - x^k\| < \varepsilon$

* Relative criteria:

(4) $\dfrac{|f(x^{k+1}) - f(x^k)|}{|f(x^k)|} < \varepsilon$

(5) $\dfrac{\|x^{k+1} - x^k\|}{\|x^k\|} < \varepsilon$

→ May work better in practice

Convergence of Gradient Methods

10/14/24

Lecture 8

Convergence Types

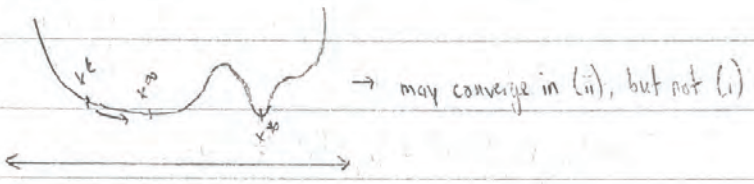
3 things to consider:

(1) What is converging?

(i) Convergence of iterates: $\|x^k - x^*\| \to 0$ as $k \to \infty$

(ii) Convergence of functions: $|f(x^k) - f(x^*)| \to 0$ as $k \to \infty$

- Generally less powerful & easier than (i)



→ may converge in (ii), but not (i)

(2) Where are we converging from? [i.e from which x^0 do we converge]

(i) Locally: The algorithm converges in some neighborhood $B_\varepsilon(x^*)$ $[\varepsilon > 0]$, but may fail to converge for starting points $x^{(0)} \notin B_\varepsilon(x^*)$

(ii) Globally: The algorithm converges from any starting point $x^{(0)} \in \Omega$

(3) How fast do we converge?

Convergence of Gradient Methods

(*) $\nabla f(x) = Qx + b$; $D^2 f(x) = Q$

For simplicity: only look at convergence for quadratic forms $f(x) = \frac{1}{2} x^T Q x + b^T x$ with $Q > 0$, $Q = Q^T$.

$\rightsquigarrow \exists$ closed-form solution:

$$\boxed{x^* = Q^{-1} b} \quad \text{(min. of } f\text{)}$$

Can consider $V: \mathbb{R}^n \to \mathbb{R}$ def. by:

$$V(x) = f(x) + \frac{1}{2}(x^*)^T Q x^*$$

$(*) = (x - x^*)^T Q (x - x^*) \Rightarrow V, f$ have same min.

Convergence of Gradient Methods (cont.)

10/14/24
Lecture 8 (cont.)

Goal: Want to show that $x^k \to x^*$ [x^k: via fixed step size GD] (Q P.D. $\Rightarrow V(x^k) \to 0$ only if $x^k \to x^*$)

→ **Observation:** Suffices to show that $V(x^k) \to V(x^*)$ [$=0$]; "$x^k \to x^*$" $\iff$ "$V(x^k) \to V(x^*)$"

Lemma: The iterative algorithm $x^{k+1} := x^k - \alpha_k g^k$ [$g^k(x) = Qx^k - b$] satisfies:

$$V(x^{k+1}) = (1-\gamma_k) V(x^k)$$

where $\gamma_k = 1$ if $g^k = 0$, otherwise $\gamma_k = \alpha_k \frac{(g^k)^T Q g^k}{(g^k)^T Q^{-1} g^k} \left(2\frac{(g^k)^T g^k}{(g^k)^T Q g^k} - \alpha \right)$.

(*) **Proof:** "Via direct computation i can find closed-form soln for α_k"

↓

Theorem: Let $\{x^k\}$ be defined as $x^{k+1} := x^k - \alpha_k g^k$, and let γ_k as in (Lemma); assume $\gamma_k > 0$.

→ Then $\{x^k\} \to x^*$ for any x^0 iff $\boxed{\sum_{k=0}^{\infty} \gamma_k = \infty}$

↳ **Note:** This independent of step size!

(*) **Proof** ($\Leftarrow$)

From (Lemma): have that $V(x^k) = \left(\prod_{i=0}^{k-1} [1-\gamma_i] \right) V(x^0)$. Assume $\gamma_k < 1$.

→ **Obs.:** $x^k \to x^* \iff V(x^k) \to V(x^*) = 0$

$$\Rightarrow \prod_{i=0}^{\infty} [1-\gamma_i] = 0 \iff \sum_{i=1}^{\infty} -\log(1-\gamma_i) = \infty$$

→ Remains to show: $\sum_{i=1}^{\infty} \gamma_i = \infty \Rightarrow \sum_{i=1}^{\infty} -\log(1-\gamma_i) = \infty$.

Observe: $\forall\, x > 0$, $\log(x) \le x - 1 \Rightarrow \log(1-\gamma_i) \le 1 - \gamma_i - 1 = -\gamma_i$

$$\Rightarrow \sum_{i=1}^{\infty} -\log(1-\gamma_i) \ge \sum_{i=1}^{\infty} \gamma_i = \infty$$

Convergence of Steepest Descent

10/14/24

Lecture 8 + Lecture 9 + Disc 5

Convergence of Steepest Descent

Know [for f, Q as before]: (1) $\lambda_{min}(Q)\|x\|^2 \le x^TQx \le \lambda_{max}(Q)\|x\|^2$

+ (2) Since $Q>0$: $\lambda_{min}(Q^{-1}) = \frac{1}{\lambda_{max}(Q)}$, $\lambda_{max}(Q^{-1}) = \frac{1}{\lambda_{min}(Q)}$

⇒ Lemma: For $Q=Q^T$, $Q>0$:

$$\frac{\lambda_{min}(Q)}{\lambda_{max}(Q)} \le \frac{(x^Tx)^2}{(x^TQx)(x^TQ^{-1}x)} \le \frac{\lambda_{max}(Q)}{\lambda_{min}(Q)}$$

2 upper & lower bounds

Thm: For steepest descent on f, $x^k \to x^*$ for any x^0.

(*) Proof: $g^k \Rightarrow x^k = x^*$; assume $g^k \neq 0$. Can solve for α_{opt} explicitly:

- Let $d^k = -g^k = -Qx^k + b$
  - → define $\Phi(\alpha) = f(x^k + \alpha d^k)$

$$= \tfrac{1}{2}(x^k + \alpha d^k)^T Q(x^k + \alpha d^k) - b^T(x^k + \alpha d^k)$$ (quadratic eqn.)

$$= \alpha^2\underbrace{\left(\tfrac{1}{2}d^kQd^k\right)}_{a_2} + \alpha\underbrace{d^{kT}(Qx-b)}_{a_1} + \underbrace{\text{const.}}_{a_0} = a_2\alpha^2 + a_1\alpha + a_0$$

→ $\alpha_{opt} = \underset{\alpha}{\text{argmin}}\,\Phi(\alpha) = -\frac{a_1}{2a_2} = -\frac{d^{kT}(Qx-b)}{d^TQd} = \boxed{\frac{g^Tg}{g^TQg}}$

→ substituting into γ_k:

$$\gamma_k = \frac{((g^k)^Tg^k)^2}{((g^k)^TQg^k)((g^k)^TQg^k)} \ge \frac{\lambda_{min}(Q)}{\lambda_{max}(Q)} > 0 \to \text{implies } \sum_k^\infty \gamma_k = \infty.$$ ▨

(*) Quadratic Forms

$f = \frac{1}{2}x^TQx + b^Tx + c$

→ $\nabla f(x) = Qx + b$ $\quad \alpha^*$ [steepest descent] $= \frac{g^Tg}{g^TQg}$, $g = Qx - b$

& $D^2f(x) = Q$ $\quad x^*$ [$Q>0$, $Q=Q^T$] $= Q^{-1}b$

(*) Q not symmetric → can substitute with $\hat{Q} = \frac{1}{2}(Q + Q^T)$

Convergence Rates of Gradient Methods

10/16/24
Lecture 9 (cont.)

Convergence of Fixed-Step-Size GD $[x^{k+1} := x^k - \alpha \nabla f(x^k),\ \alpha > 0]$

Thm: For fixed-step-size GD, $x^k \to x^*$ for any x^0 iff: $\boxed{0 < \alpha < \frac{2}{\lambda_{max}(Q)}}$

[2] (*) For quadratic form f

(*) Proof ($\Leftarrow$)

By Rayleigh's inequality, have that: (1) $\lambda_{min}(Q)(g^k)^T g^k \le (g^k)^T Q g^k \le \lambda_{max}(g^k)^T g^k$

(2) $(g^k)^T Q^{-1} g^k \le \frac{1}{\lambda_{min}(Q)} (g^k)^T g^k$

→ Substituting into γ_k:

$$\gamma_k \ge \alpha \left(\lambda_{min}(Q)\right)^2 \left(\frac{2}{\lambda_{max}(Q)} - \alpha\right) > 0 \longrightarrow \gamma_k > 0 \ \forall k \Rightarrow \sum_k^\infty \gamma_k = \infty.$$ (by assumption)

Convergence Rates

Similar to convergence types, have many ways to talk about convergence rates; e.g.:

- Convergence rates for iterates
- Convergence rates for functions (trickier)

(*) **Convergence Rate of Steepest Descent** ($x^* = Q^{-1}b$)

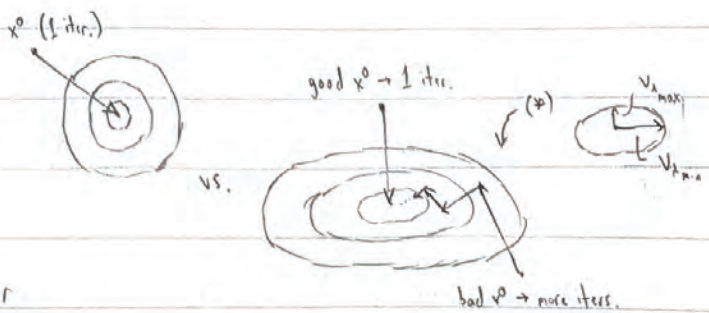
Let $f(x) = \frac{1}{2} x^T Q x + b^T x + c$ [as before], $V(x) = f(x) + \frac{1}{2}(x^*)^T Q x^*$

→ $V(x^{k+1}) \le \frac{\lambda_{max}(Q) - \lambda_{min}(Q)}{\lambda_{max}(Q)} V(x^k)$; define $\boxed{r := \frac{\lambda_{max}(Q)}{\lambda_{min}(Q)} \ge 1}$ ← important for convergence rate

$\Rightarrow \underline{0 \le V(x^k) \le \left(1 - \tfrac{1}{r}\right)^k V(x^0)}$.

Looking at r:

- $r = 1$ [circular level sets] => converges in 1 iteration
- $r > 1$ [eccentric level sets] => may be slower



Orders of Convergence

10/16/24

Lecture 9

+Lecture 10

Orders of Convergence

Often convenient to talk about order of convergence [of iterates, e.g.]:

(*) $\lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|^q} = 0$ $\Rightarrow$ order $> q$

- Order-p convergence: $\boxed{0 < \lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|^p} < \infty}$

- Order-∞ convergence: $\boxed{\lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|^p} = 0}$ $\forall\, p$

(*) Important Cases

- $p=1 \rightarrow$ sublinear if $\lim_{k\to\infty} \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|} = 1$, linear if < 1
- $p>1 \rightarrow$ superlinear
- $p=2 \rightarrow$ quadratic

(*) Ex:

- Quadratic $f \rightarrow$ worst-case [for GD] rate/order $= 1$

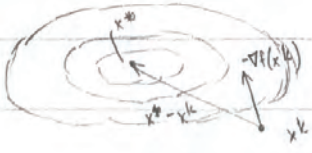
(*) Ex: $f:\mathbb{R}\to\mathbb{R}$, $f(x) = x^2 - x^3/3$, $\alpha = 1/2$, $x^0 = 1$ $\rightarrow$ converges to $x=0$

- Rate: $x^{k+1} = x^k - \alpha f'(x^k) = \frac{1}{2}(x^k)^2$

$\rightarrow x^k = \left(\frac{1}{2}\right)^{2^k-1} \longrightarrow \frac{|x^{k+1}|}{|x^k|^2} = \frac{1}{2}$ [order 2]

Improving on Gradient Methods

Gradient methods use $-\nabla f(x)$ as search dir. $\rightarrow$ saw shortcomings; e.g. for x^* w/ $\frac{\lambda_{max}(\nabla^2 f(x))}{\lambda_{min}(\nabla^2 f(x))} \gg 1$, may have very slow convergence (from analysis of steepest descent for quadratic form f)



$\rightarrow$ Goal: Transform $-\nabla f(x)$ s.t. it points more toward x^*

Newton's Method for $\mathbb{R}^n$

10/18/24

Lecture 10

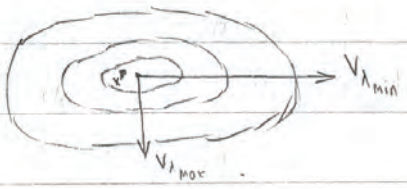
Newton's Method in $\mathbb{R}^n$

Observations:

(*) Q symmetric [$\Rightarrow$ eigenvectors span $\mathbb{R}^n$]

(1) Given quadratic $f: \mathbb{R}^2 \to \mathbb{R}^2$, $f(x) = \frac{1}{2} x^T Q x$ → can always express $\nabla f(x)$ in terms of the eigenvectors $v_{\lambda_{min}}, v_{\lambda_{max}}$ of Q (for any x):

$$\boxed{\nabla f(x) = c_1 v_{\lambda_{min}} + c_2 v_{\lambda_{max}}}$$



→ Can multiply by Q^{-1} to point $-\nabla f(x)$ towards x^*:

$$Q^{-1}\nabla f(x) = \frac{c_1}{\lambda_{min}(Q)} v_{min} + \frac{c_2}{\lambda_{max}(Q)} v_{max} \quad ; \quad \text{(i) } c_1 = \lambda_{min}(Q) \cdot v_{min}^T x, \quad \text{(ii) } c_2 = \lambda_{max}(Q) \cdot v_{max}^T x$$

$$\Rightarrow Q^{-1}\nabla f(x) = x \Rightarrow x^{(1)} = x^{(0)} - Q^{-1}\nabla f(x^{(0)}) = 0 = x^* \quad \text{↳ converge in 1 iter.}$$

→ (2) For general f, can use 2nd-order to info → can treat $D^2 f(x)$ as Q.

Approach: Use Taylor expansion to create 2nd-order/quadratic approximation of $f: \mathbb{R}^2 \to \mathbb{R}$:

$$f(x) \approx f(x^{(0)}) + (x - x^{(0)})^T \nabla f(x^{(0)}) + \frac{1}{2}(x - x^{(0)}) D^2 f(x^{(0)})(x - x^{(0)}) =: q(x)$$

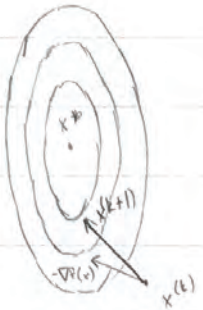
→ Solving for $\nabla q(x) = 0$:

(assuming $D^2 f(x^{(0)})$ is invertible)

$$0 = \nabla f(x^{(0)}) + D^2 f(x^{(0)})(x - x^{(0)}) \Rightarrow \boxed{x = x^{(0)} - \left(D^2 f(x^{(0)})\right)^{-1} \nabla f(x^{(0)})}$$

→ **Newton's Method [$\mathbb{R}^n$]:**

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



⟶ converges faster than pure GD, provided $D^2 f(x) > 0$ [important]

Newton's Method for $\mathbb{R}^n$ (cont.)

10/18/24

Lecture 10 + Lecture 11

Remark: Similar to the 1D case, can use Newton's method as a general means for finding 0s:

(1) Solve $D(g^k)d^k = -g(x^k)$ for d^k $\rightarrow$ (2) Set $x^{k+1} := x^k + d^k$.

Limitations of Newton's method:

(i) Needs efficient way to solve for/invert $n \times n$ matrices $\rightarrow$ may be limiting in higher dimensions

(ii) Requires $D^2 f(x) > 0$; otherwise, $\nabla f(x)$ may fail to be a descent direction

(iii) Even if $D^2 f(x) > 0$, no guarantee that $f(x^{k+1}) < f(x^k)$

$\rightarrow$ **Q's:** 1. Does Newton's method converge?

2. At what rate does it converge?

3. Can we make Newton's method more robust w.r.t. initial $x^{(0)}$?

Analysis of Newton's Method

No global convergence with Newton's method; want to show: (i) Local convergence [1]

+ (ii) Newton's method converges faster than gradient methods [2]

+ (iii) Can use 1D line search to stabilize from initial conditions

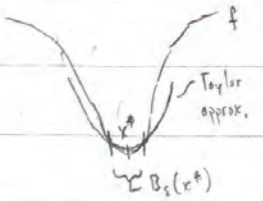
> **Thm:** Suppose $f \in C^3$ and $x^* \in \mathbb{R}^n$ s.t. $\nabla f(x^*) = 0$ & $D^2 f(x^*)$ is invertible; then $\exists$ $\varepsilon > 0$ s.t. $\forall$ $x^0 \in B_\varepsilon(x^*)$ sufficiently close to x^*, Newton's method is well-defined (i.e. $D^2 f(x)^{-1}$ exists) and converges to x^* with order of at least 2.

(*) **Rem:** Gradient methods converge w/ order of most 1 $\rightarrow$ Newton's method is faster

Convergence of Newton's Method in $\mathbb{R}^n$

10/21/24
Lecture 11
(cont.)

(*) Proof of Theorem



First: can find a local bound for $\|D^2f(x_0)(x-x_0)-\nabla f(x_0)\|$, $\|(D^2f(x))^{-1}\|$

→ via Taylor expansion on ∇f at x^0:

$$\nabla f(x) = \nabla f(x^0) + D^2f(x^0)(x-x_0) + O(\|x-x_0\|^2)$$

(using that $f \in C^3$)

$$\rightarrow \nabla f(x) - \nabla f(x^0) - D^2f(x^0)(x-x_0) \le c_1\|x-x_0\|^2 \quad [\text{for some } c_1 \in \mathbb{R}]$$

+ from regularity: for $x \in B_\varepsilon(x^*)$, have that $\|(D^2f(x))^{-1}\| \le c_2$ [for some $c_2 \in \mathbb{R}$]

Substituting $x^0 \to x^*$: $\nabla f(x) - D^2f(x)(x-x^*) = O(\|x-x^*\|^2)$

+ also:

$$\begin{aligned}
\|x^{(1)} - x^*\| &= \|x^{(0)} - (D^2f(x))^{-1}\nabla f(x^{(0)}) - x^*\| \\
&= \|-(D^2f(x))^{-1}\left(-D^2f(x^{(0)})(x^* - x^{(0)}) + \nabla f(x^{(0)})\right)\| \\
&\le \|(D^2f(x))^{-1}\| \cdot \|D^2f(x^{(0)})(x^* - x^{(0)}) + \nabla f(x^{(0)})\| \\
&\le c_2 \cdot c_1\|x^{(0)} - x^*\|^2
\end{aligned}$$

Assume $x^{(0)}$ is such that $\|x^0 - x^*\| \le \frac{\alpha}{c_1c_2}$, $\alpha \in (0,1)$

$\rightarrow c_2c_1\|x^{(0)} - x^*\|^2 \le \alpha\|x^{(0)} - x^*\|$

$\rightarrow$ by induction: $\|x^{k+1} - x^*\| \le \alpha\|x^k - x^*\| \quad \forall k$

$\rightarrow \lim_{k\to\infty}\|x^k - x^*\| = 0 \quad [= \alpha^k\|x^{(0)} - x^*\|]$

+ looking at convergence rate:

$$\lim_{k\to\infty}\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} \le c_1c_2 \in (0,\infty) \rightarrow \text{converges w/ order 2.}$$

Modifications of Newton's Method

10/21/24
Lecture 11
+ Lecture 12

Thm: Let $\{x^k\}$ be the sequence generated by Newton's method for some given $f: \mathbb{R} \to \mathbb{R}$. If $\nabla f(x^k) \neq 0$ & $\nabla^2 f(x) > 0 \ \forall k$, then any search dir. $d^k := -(\nabla^2 f(x))^{-1} \nabla f(x^k)$ is a descent direction [of f].

Can use (1D) line search to find step size $\alpha \to$ "Modified Newton": $x^{k+1} = x^k - \alpha^k d^k$,
$\alpha^k = \operatorname{argmin}_{\alpha \geq 0} f(x^k - \alpha d^k)$

(*) In practice: start with gradient methods
→ once improvement slows, switch to Newton's method

(*) In higher dims.: require more iters. + radius of convergence for Newton's decreases exponentially

Modifications of Newton's Method

Previously: Newton's converges to a local min. if $\nabla^2 f(x) > 0$ + step size $\alpha \leq 1$ → can we relax this?
→ Look at: $x^{k+1} = x^k - (\nabla^2 f(x) - \mu I)^{-1} \nabla f(x^k)$

Claim: This (i) "solves" positive-definiteness and (ii) acts as step size.

(*) "Proof".

Let $\lambda_1, \dots, \lambda_n$ be eigenvalues of $\nabla^2 f(x)$ → $(\nabla^2 f(x) + \mu I)$ has eigenvalues $\lambda_i + \mu$.
→ Observe: Can pick $\mu > -\min(\lambda_{\min}, 0)$ s.t. $(\nabla^2 f(x) + \mu I) > 0$.
+ If μ small enough, can still expect new x^{k+1} to be close to original x^{k+1} (from Newton's method)

+ Observation: $\nabla^2 f(x)$ symmetric → can write $\nabla^2 f(x) = V \Lambda V^T$, $V = [v_1 \dots v_n]$, $\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$
→ Looking at d^k: $x^{k+1} - x^k = d^k := \nabla^2 f(x^k)^{-1} \nabla f(x^k) = V^T \Lambda^{-1} V \nabla f(x^k)$

$$\Rightarrow \boxed{\|d^k\| \leq \frac{1}{\min(\lambda_i)} \|\nabla f(x^k)\|} \longrightarrow \text{relaxed Newton: } \boxed{\|d^k\| \leq \frac{1}{\min(\lambda_i) + \mu} \|\nabla f(x^k)\|}$$

(*) namely: as $\mu \to \infty$, $\|d^k\| \to 0$; acts as a form of step size

Modifications of Newton's Method (cont.)

10/23/24

Lecture 12

Levenberg-Marquardt Algorithm

For every iteration k, want to find μ_k s.t. $f(x^{k+1}) < f(x^k)$:

$$x^{k+1} := x^k - \left(D^2 f(x^k) - \mu_k I\right)^{-1} \nabla f(x^k)$$

(*) Can view as a combination of gradient descent & Newton's method

- In particular: for μ large, μI dominates & method mimics GD w/ small step size

Newton's Method for Nonlinear Least Squares

Besides Levenberg-Marquardt, can make other modifications for specific functions $f: \mathbb{R}^n \to \mathbb{R}$

→ One case: $f(x) = \sum_{i=0}^{m} (r_i(x))^2$, $r_i: \mathbb{R}^n \to \mathbb{R}$ [Nonlinear least squares]

(*) Ex: $r_i: \mathbb{R}^3 \to \mathbb{R}$, $r_i(A, \omega, \phi) = y_i - A\sin(\omega t + \phi)$, $\{y_i\} \subseteq \mathbb{R}$, $\{t_i\} \subseteq \mathbb{R}$

↳ Write problem as: $f(x) = r^T(x) r(x)$, $r: \mathbb{R}^n \to \mathbb{R}^m$, $r(x) = \langle r_1(x), \dots, r_m(x) \rangle$

For Newton's method, need (i) gradient + (ii) Hessian

(i) $\left(\nabla f(x)\right)_j = \frac{\partial}{\partial x_j} f(x) = 2 \sum_{i=0}^{m} r_i(x) \frac{\partial}{\partial x_j} r_i(x)$

(ii) $D^2 f(x) = 2 \sum_{i=0}^{m} r_i(x) D^2 r_i(x) + 2 Dr(x)^T Dr(x)$

- $r_i(x) D^2 r_i(x)$: in practice, difficult to compute; may opt to throw out, e.g. if we expect $r_i(x^k) \approx 0$
- $Dr(x)^T Dr(x)$: P.S.D.

(*) In some cases: can approximate/estimate $D^2 f(x)$ with Q s.t. $\|D^2 f(x) - Q\| < \varepsilon$ for some $\varepsilon > 0$

→ **Gauss-Newton Method**

$$x^{k+1} := x^k - \left(Dr(x^k)^T Dr(x^k)\right)^{-1} Dr(x^k) r(x^k)$$

2 In practice: can add μI term (Levenberg-Marquardt)

(*) Remarks:

- G-N doesn't require 2nd derivative (easier to compute), works even if 2nd derivative $= 0$
- G-N converges superlinearly if similar convergence to Newton's method, radius of convergence dep. on $\frac{\lambda_{max}}{\lambda_{min}}$

Conjugate Direction Methods

10/25/24
Lecture 13 + Disc. 6

Previously, saw iterative search directions $d^k = -\nabla f(x^k)$ [GD], $-(\nabla^2 f(x^k))^{-1}\nabla f(x^k)$ [Newton's method]

↝ conjugate direction methods as "intermediate" methods between GD, Newton's

(Newton's > C.Dir. > GD, but no matrix inversion needed + Hessian eval. not required)

For quadratic $f(x) = \frac{1}{2}x^TQx - x^Tb$, $Q = Q^T > 0$:

> **Def:** A set of directions $d^1, d^2, \dots, d^n$ are said to be Q-conjugate if, $\forall i \neq j$, $(d^i)^T Q d^j = 0$.

→ Notice: Q-conjugate directions are always linearly independent.

(Find $d^1, \dots, d^k$ Q-conj: via Gram-Schmidt, e.g.)

The Conjugate Direction Algorithm: Given Q-conjugate directions $d^1, \dots, d^k$:

$$x^{k+1} := x^k + \alpha_k d^k \quad \text{where} \quad g^k = Qx^k - b \ (= \nabla f(x^k)) \longrightarrow \alpha_k = \frac{(g^k)^T d^k}{(g^k)^T Q g^k}$$

[stop if $g^{k+1} = 0$]

Thm: For any x^0, the basic conjugate direction algorithm (given n Q-conjugate directions, where $x^k \in \mathbb{R}^n$) converges to $x^* = Q^{-1}b$ in [at most] n steps, i.e. $x^n = x^*$.

Remark: α_k is similar to in steepest descent, but conj dir satisfies $f(x^{k+1}) = \min_{\alpha_1, \dots, \alpha_n} f(x^0 + \sum_{i=1}^{k} \alpha_i d^i)$,

i.e. x^{k+1} is the best possible estimate for the min x^* within subspace spanned by $d^1, d^2, \dots, d^k$

Q: How to find Q-conj. directions?

(*) popular for numerical computing

> **The Conjugate Gradient Method:** Taking g^k, α^k, x^k as before, find directions d^k by solving:
>
> $$\beta_k = \frac{(g^{k+1})^T Q d^k}{(d^k)^T Q d^k} \longrightarrow d^{k+1} := -g^{k+1} + \beta_k d^k$$
>
> [d^k's Q-conjugate]

(*) In practice, may be near x^* even before n iterations; # iterations needed depends on $r := \frac{\lambda_{max}(Q)}{\lambda_{min}(Q)}$

Quasi-Newton Methods

10/28/24

Lecture 14

Quasi-Newton methods - a distinct class of methods, related to Newton's method

→ Basic outline: For $f: \mathbb{R}^n \to \mathbb{R}$, follow:

$$\boxed{x^{k+1} := x^k - \alpha_k H_k \nabla f(x^k)} \quad \text{for some } H_k \in \mathbb{R}^{n \times n}, \; H_k = H_k^T, \quad \alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x^k + \alpha d^k)$$

- Newton's method: $H_k = (D^2 f(x^k))^{-1}$ → Quasi-Newton: approximate inverse Hessian w/ H_k symmetric

→ Observe: If $H_k > 0$, obtain a descent direction (proof via Taylor's theorem)

Approximating the Inverse Hessian

Q: How to construct $H_k > 0$ without finding the explicit inverse of the Hessian?

- Can take successive approximations $H_0, H_1, H_2, \ldots$ of $(D^2 f(x))^{-1}$

→ Goal: Want to find a general condition these H_k's must satisfy

Notation: $\underbrace{\nabla f(x^{k+1}) - \nabla f(x^k)}_{\Delta g^k} = Q\underbrace{(x^{k+1} - x^k)}_{\Delta x^k}$ [for f quadratic] $\longrightarrow \Delta g^k = Q \Delta x^k$

Notice: For quadratic f, $\forall k$, satisfies $Q^{-1} \Delta g^i = \Delta x^i \;\; \forall i = 1, \ldots, k$

→ can impose a similar constraint for quasi-Newton H_k's:

for every k: $\boxed{H_{k+1} \Delta g^i = \Delta x^i}$ for $i = 1, \ldots, k$

After n steps ($k = n-1$), obtain n equations $H_n \Delta g^i = \Delta x^i \;\; [i = 1, \ldots, n]$

→ can write as single system: $\boxed{H_n \Delta G^n = \Delta X^n}$ $\xrightarrow{(\text{if } \Delta G^n \text{ nonsingular})}$ $\boxed{H_n = \Delta X^n (\Delta G^n)^{-1}}$ (unique solution)

Solution is unique & Q^{-1} a solution $\Rightarrow H_n = Q^{-1}$; process yields Q^{-1} after n iterations

→ on $(n+1)^{\text{st}}$ iteration: converges to x^* in a single step (for quadratic f)

Quasi-Newton Methods (cont.)

10/28/24

Lecture 14 + Lecture 15

Quasi-Newton methods: $x^{k+1} := x^k + \alpha_k d^k$, where:

$$\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}} f(x^k + \alpha d^k) \quad \text{and} \quad d^k = -H_k \nabla x^k$$

↪ for H satisfying: (i) $H_k = H_k^T > 0$

(ii) $H_{k+1} \Delta g^i = \Delta x^i$ for $i = 1, \dots, k$

Remarks: (i) Quasi-Newton methods are actually conjugate-direction methods: d^k are Q-conjugate for quadratic f

(ii) For $k < n$, no unique solution for H_{k+1}; need to pick "well-performing" H_k

→ Q: How to choose H_k that is stable and robust?

(1) Rank-One Correction / SR1 Algorithm

Approach: Keep adding "degrees of freedom":

$$H_{k+1} := H_k + a_k z^k (z^k)^T \quad \text{where } a_k \in \mathbb{R},\ z^k \in \mathbb{R}^n$$

(Observe: $\operatorname{rank}[z^k (z^k)^T] = 1$)

→ From constraint, derive formula:

$$a_k z^k (z^k)^T = \frac{(\Delta x^k - H_k g^k)(\Delta x^k - H_k \Delta g^k)^T}{(\Delta g^k)^T (\Delta x^k - H_k \Delta g^k)}$$

(*) Remarks:

1. Denominator may be negative → before n^{th} iteration, may have negative eigenvalues [non-PD H_k]
2. Denominator may be close to 0 → alg. may be numerically unstable

(2) Rank-Two Correction / DFP Algorithm

Given $x^0 \in \mathbb{R}$ and H_0 any symmetric PD matrix (e.g. $I_{n \times n}$), define (for $k = 1, 2, \dots$):

(i) d^k, α_k, g^{k+1}, x^{k+1} as before ($g^0 = \nabla f(x^0)$; stop if $g^k = 0$)

(ii) H_k:

$$H_{k+1} := H_k + \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta g^k} - \frac{H_k \Delta g^k (H_k \Delta g^k)^T}{(\Delta g^k)^T H_k \Delta g^k}$$

Quasi-Newton Methods (cont.)

10/30/24
Lecture 15 (cont.)

(*) Remarks (DFP Algorithm)

1. Unlike rank-1 correction, the DFP matrix H_{k+1} is P.D. if H_k is P.D.
2. In practice, DFP can get "stuck" on larger non-quadratic problems, where H_k become singular (or nearly singular) → yet another algorithm

New approach: rather than approximating $(D^2 f(x))^{-1}$ directly, approximate $D^2 f(x)$ instead and then take inverse (using tricks to compute it efficiently)

↓

(3) BFGS Algorithm

Taking x^k, d^k, α_k, g^{k+1} as before, define:

$$B_{k+1} = B_k - \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta g^k} - \frac{B_k \Delta g^k (B_k \Delta g^k)^T}{(\Delta g^k)^T B_k \Delta g^k} \longrightarrow H_{k+1} = (B_{k+1})^{-1}$$

Lemma (Sherman-Morris formula): Let $A \in \mathbb{R}^{n \times n}$ nonsingular & $u, v \in \mathbb{R}^n$ s.t. $1 + v^T A u \neq 0$; then:

$$(A + u^T v)^{-1} = A^{-1} - \frac{(A^{-1}u)(v^T A^{-1})}{1 + v^T A u}$$

→ Applying to BFGS: (twice)

$$H_{k+1} = H_k + \left(1 + \frac{(\Delta g^k)^T H_k \Delta g^k}{(\Delta g^k)^T \Delta x^k}\right) \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta g^k} - \frac{H_k \Delta g^k (\Delta x^k)^T + (H_k \Delta g^k (\Delta x^k)^T)^T}{(\Delta g^k)^T \Delta x^k}$$

(*) Unlike DFP, is fairly robust to inexact line search

(*) Remarks

- Quasi-Newton methods do not need modification for non-quadratic f (unlike conjugate gradient "CG")
- Similar to CG, no guarantee of convergence within n iters for non-quadratic f
- In practice, quasi-Newton methods benefit from "restarting" from new H_0 occasionally (e.g. every $n/n+1$ iters.)

Solving Linear Systems

11/6/24

Lecture 16

Linear systems: Find $x \in \mathbb{R}^n$ s.t. $\boxed{Ax = b}$ for $A \in \mathbb{R}^{m \times n}$ $[m \geq n]$, $b \in \mathbb{R}^m$, $\text{rank}(A) = n$ ← special kind of optimization prob., common in applied probs.

→ if $b \in R(A)$, can solve; otherwise, $b \notin R(A)$ [system inconsistent] → no exact solution (happens in practice)

Goal: Pick $x \in \mathbb{R}^n$ minimizing least-square error: $\boxed{x^* = \underset{x \in \mathbb{R}^n}{\text{argmin}} \|Ax - b\|^2}$

- **Lemma:** Let $A \in \mathbb{R}^{m \times n}$, $m \geq n$ → $\text{rank}(A) = n \iff \text{rank}(A^TA) = n$ [A^TA is invertible]

→ **Thm:** The unique $x^* \in \mathbb{R}^n$ minimizing $\|Ax - b\|^2$ is given by solving:

$$A^TAx^* = A^Tb \longrightarrow \boxed{x^* = (A^TA)^{-1}A^Tb}$$ [least squares solution]

(*) Proof ~ (*) Alt. proof: Via FONC, set $\nabla \|Ax - b\|^2 = 0$

Observe: for $x \in \mathbb{R}^n$, $\|Ax - b\|^2 = \|A(x - x^*) + A(x^* - b)\|^2$

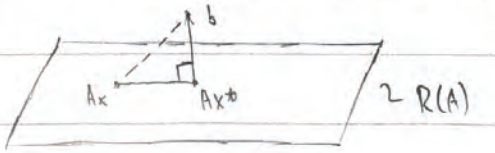
$$= \|\underbrace{A(x - x^*)}_{(1)}\|^2 + \|\underbrace{A(x^* - b)}_{(2)}\|^2 + 2\underbrace{\left(A(x - x^*)\right)^T\left(A(x^* - b)\right)}_{(3)}$$

→ Can show (3) is 0: $\left(A(x - x^*)\right)^T\left(A(x^* - b)\right) = (x - x^*)^TA^T\left(A(A^TA)^{-1}A^Tb - b\right)$

$$= (x - x^*)^T\left(A^TA(A^TA)^{-1}A^Tb - A^Tb\right)$$

$$= (x - x^*)\left(A^Tb - A^Tb\right) = 0$$

$\Rightarrow \|Ax - b\|^2 = \|A(x - x^*)\|^2 + \|A(x^* - b)\|^2$, minimized by $x = x^*$. ▨



Recursive Least Squares (RLS)

New problem: Assume we have solution x^0 for $A_0x = b_0$, but new measurements A_1, b_1 come in

→ Want to find least-squares soln. for $\boxed{\begin{bmatrix} A_0 \\ A_1 \end{bmatrix} x = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}}$

Goal: Want to update $x^0 = (A_0^TA_0)^{-1}A_0^Tb_0$ to find $x^1 = \underset{x \in \mathbb{R}^n}{\text{argmin}} \left\| \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} x - \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right\|$

Recursive Least Squares

11/6/24

Lecture 16 (cont.)

Recursive Least Squares (cont.)

(formula for linear system)

Observe: Can write $x' = G_1^{-1}\begin{bmatrix} A_0 \\ A_1 \end{bmatrix}^T \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$, where $G_1 = \begin{bmatrix} A_0 \\ A_1 \end{bmatrix}^T \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} = G_0 + A_1^T A_1$

$\rightarrow$ $x' = x^0 + G_1^{-1} A_1^T (G' - A_1 x_0)$ [update rule]

Recall: With quasi-Newton methods, saw that $\exists$ a formula for G_1^{-1} if $A_1^T A_1$ is rank 1

$\rightarrow$ **Lemma (Sherman-Morrison-Woodbury):** Let A be nonsingular, U & V matrices s.t. $1 + VA^{-1}U \neq 0$; then:

$$\boxed{(A + UV)^{-1} = A^{-1} - (A^{-1}U)(1 + VA^{-1}U)(VA^{-1})}$$

↪ general form for Sherman-Morrison

$\rightarrow$ Replacing G_k^{-1} with P_k, obtain general RLS algorithm:

$$\boxed{P_{k+1} = P_k A_{k+1}^T (1 + A_{k+1} P_k A_{k+1}^T) A_{k+1} P_k}$$

$$\longrightarrow \quad \boxed{x_{k+1} = x^k + P_{k+1} A_{k+1}^T (b^{k+1} - A_{k+1} x^k)}$$

(*) RLS (Rank-1 Update)

If update A_{k+1} is rank 1 (i.e. A_{k+1} is a singular vector a_{k+1}), can use Sherman-Morrison:

$$\boxed{P_{k+1} = P_k - \frac{P_k a_{k+1} a_{k+1}^T P_k^T}{1 + a_{k+1}^T P_k a_{k+1}}} \longrightarrow \boxed{x_{k+1} = x_k + P_{k+1} a_{k+1} (b_{k+1} - a_{k+1}^T x_k)}$$

(*) In applied settings (e.g. ML), use nonlinear diffeomorphisms Φ to convert linear transformations Ax into nonlinear transformations $A\Phi(x)$

Minimum-Norm Solutions

11/8/24

Lecture 17

Solving Linear Systems (cont.)

Alternate case: Want to solve $Ax = b$, where $A \in \mathbb{R}^{m \times n}$, $\operatorname{rank} A = m$, $m \leq n$

→ infinitely many solutions exist; can choose to consider soln. closest to the origin: $\min_x \|x\|$ s.t. $Ax = b$

Thm: This problem has unique solution given by $\boxed{x^* = A^T(AA^T)^{-1}b}$

(*) Proof: Observe: $\|x\|^2 = \|x - x^* + x^*\|^2 = \left((x-x^*) + x^*\right)^T\left((x-x^*) + x^*\right)$

$= \|x - x^*\|^2 + \|x^*\|^2 + 2(x^*)^T(x - x^*)$.

→ Can show that final term disappears:

$(x^*)^T(x - x^*) = \left(A^T(AA^T)^{-1}b\right)^T\left(x - A^T(AA^T)^{-1}b\right)$

$= b^T(AA^T)^{-1}\left(\underbrace{Ax}_{b} - \underbrace{AA^T(AA^T)^{-1}b}_{b}\right) = 0.$ ▨

If m, n large, we want to avoid having to invert $AA^T \in \mathbb{R}^{m \times m}$ → can find x^* w/o solving any linear systems:

(*) Kaczmarz's Algorithm

Let a_j^T denote j^{th} row of A, b_j = j^{th} component of b + let $\mu \in (0,2)$:

→ Algorithm: For $i = 0, 1, \ldots$:

For $j = 1, \ldots, m$:

$$\boxed{x^{(im+j)} = x^{(im+j-1)} + \mu\left(b_j - a_j^T x^{(im+j-1)}\right)\frac{a_j}{a_j^T a_j}}$$

Thm: For Kaczmarz's algorithm with $x^0 = 0$, $x^k \to x^* = A^T(AA^T)^{-1}b$ as $k \to \infty$.

Rmks: (i) For $x^0 \neq 0$, Kaczmarz's algorithm will converge to the point $x \in \{Ax = b\}$ (linear subspace) minimizing $\|x - x_0\|$

(ii) This is useful in data science for obtaining solns efficiently, esp if more variables [n] than equations [m]

The Moore-Penrose Inverse

11/8/24

Lecture 17 (cont.)

Solving Linear Equations (General case)

General case: solve $Ax = b$ for $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = r \leq \min\{m, n\}$

→ via pseudo-inverses (specifically: Moore-Penrose inverse)

Lemma (Full-rank factorization): Let $A \in \mathbb{R}^{m \times n}$ & $\text{rank}(A) = r$; then $\exists$ matrices $B \in \mathbb{R}^{m \times r}$, $C \in \mathbb{R}^{r \times n}$ s.t. $\boxed{A = BC}$, $[\text{rank}(B) = \text{rank}(C) = r]$

- **Def:** Given $A \in \mathbb{R}^{m \times n}$, a matrix $A^+ \in \mathbb{R}^{n \times m}$ is called a pseudo-inverse of A if:

$$\boxed{AA^+A = A}$$

and $\exists$ matrices $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{n \times m}$ satisfying: $\boxed{A^+ = UA^T}$, $\boxed{A^+ = A^TV}$.

Thm: The pseudo-inverse always exists & is unique for any matrix A.

(*) **Proof (Existence):**

Let $A = BC$ be the full-rank factorization of A ($B \in \mathbb{R}^{m \times r}$, $C \in \mathbb{R}^{r \times n}$)

→ have that $A^+ = C^+B^+$, where $B^+ = (B^TB)^{-1}B^T$ and $C^+ = C^T(CC^T)^{-1}$

↳ (*) Corollaries: $(A^T)^+ = (A^+)^T$, $(A^+)^+ = A$.

→ General soln: $\boxed{x^* = A^+b}$
- Minimizes $\|Ax^* - b\|$ & $\|x^*\|$

(*) **Remarks:**

- If $n = m = \text{rank}(A)$, have that $A^+ = A^{-1}$ w/ $U = (A^TA)^{-1} = A^{-1}(A^T)^{-1}$, $V = (AA^T)^{-1}$
- Can verify that:

(i) $m \geq n$, $\text{rank}(A) = n$ → $A^+ = (A^TA)^{-1}A^T$ (RLS soln) & $A^+A = I_n$ [A^+ is a "left pseudo-inverse"]

(ii) $m \geq n$, $\text{rank}(A) = m$ → $A^+ = A^T(AA^T)^{-1}$ [min-norm soln] & $AA^+ = I_m$ [A^+ a right pseudoinverse]

- Can interpret U, V conditions as ensuring that each row of A^+ is a linear combination of the rows of A^T, and likewise for columns

Intro to Linear Programming

11/13/24

Lecture 18

Linear programs: A special class of constrained optimization problems (common in real-world settings)

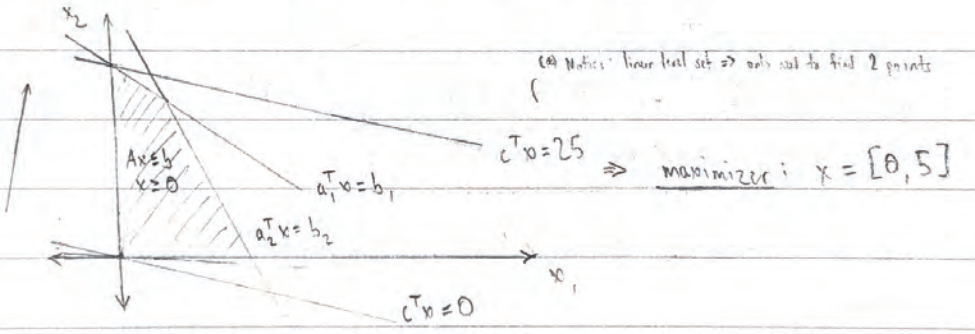
(*) Ex: $\min_{x \in \mathbb{R}^n} c^T x$ s.t. $Ax = b$, $x \geq 0$ where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$ ["$x \geq 0$" $\Leftrightarrow$ all elements of $x \geq 0$]

Two-Dimensional Linear Programs

Can easily visualize properties of linear programs in $\mathbb{R}^2$ ($\rightarrow$ use to help solve in higher-dims.)

(*) Ex: Consider the LP $\max_x c^T x$ s.t. $Ax \leq b$, $x \geq 0$, where $c = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$, $A = \begin{bmatrix} 5 & 6 \\ 3 & 2 \end{bmatrix}$, $b = \begin{bmatrix} 30 \\ 12 \end{bmatrix}$

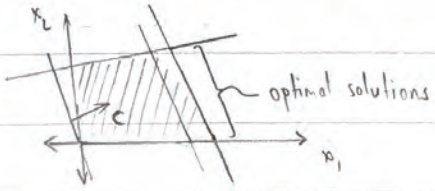
$\rightarrow$ Can draw level sets $\{x \in \mathbb{R}^2 : c^T x = \alpha\}$ of $c^T x$:



(*) Notice: linear level set $\Rightarrow$ only need to find 2 points

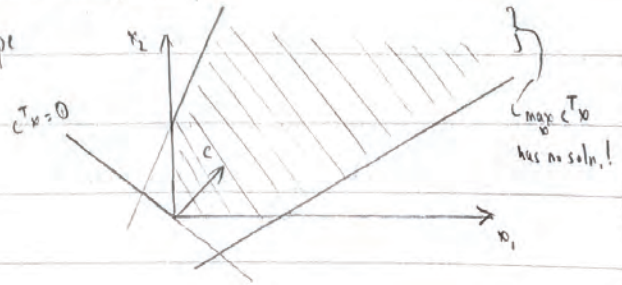
$\Rightarrow$ maximizer: $x = [0, 5]$

(*) Note: In some cases, a constraint may align perfectly with level sets of $c^T x$ $\rightarrow$ infinitely many solns.



General LPs: In general, LPs are optimization problems over convex polytopes

- Solutions may lie on faces, edges, vertices, etc of polytope
- Note: Solutions do not have to exist!



Basic Solutions to LPs

11/15/24

Lecture 19

Linear Programming

Standard form for linear programs: $\boxed{\min c^T x \text{ s.t. } Ax = b,\ x \geq 0}$

+ typically also assume [for $A \in \mathbb{R}^{m \times n}$]: $m < n$, $\text{rank}(A) = m$, $b \geq 0$

Theorems/strategies for LPs typically stated for standard form → want to be able to rewrite general LPs in standard form [only equality, ≥ 0 constraints in the above form]

(*) Ex:

(i) $\min c^T x$ s.t. ~~$Ax \geq b$~~, $x \geq 0$ → for each row, introduce "surplus variables" y_i: $A_{i1}x_1 + \dots + A_{in}x_n - y_i = b_i$, $y_i \geq 0$

→ New problem (std. form): $\min c^T x$ s.t. $\begin{bmatrix} A & (-I_n) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = b$, $\begin{bmatrix} x \\ y \end{bmatrix} \geq 0$

Notice: This does not change the minimizer! [important]

(**) For $Ax \leq b$: use "slack variables" $y_i \geq 0$, $\begin{bmatrix} A & (+I_m) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = b$

(ii) $\max\ x_2 - x_1$ s.t. $3x_1 = x_2 - 5$, $|x_2| \leq 2$, $x_1 \leq 0$

1. max → min: multiply $c^T x$ by -1
2. remove 1-1 constraint: convert to linear constraints $x_2 \leq 2$, $x_2 \geq -2$
3. flip x_1 ineq: replace w/ $\bar{x}_1 = -x_1$
4. introduce slack vars x_3, x_4

→ $\min\ -\bar{x}_1 - x_2$ s.t. $-3\bar{x}_1 = x_2 - 5$, $x_2 + x_3 = 2$, $-x_2 + x_4 = 2$, $\bar{x}_1, x_3, x_4 \geq 0$

Want constraint [≥ 0] on all vars, incl. x_2 → replace x_2 w/ u, v s.t. $x_2 = u - v$:

→ **Final problem** (std form): $\min -\bar{x}_1 - (u - v)$ s.t. $-3\bar{x}_1 - (u - v) = -5$, $(u - v) + x_3 = 2$, $-(u - v) + x_4 = 2$ and $\bar{x}_1, x_3, x_4, u, v \geq 0$

Basic Solutions to LPs

Reminder: By assump., assumed $\text{rank}(A) = m$ → more vars. than constraints (can find set of m lin indep cols. of A)

Let $B \in \mathbb{R}^{m \times m}$ be constructed from any m linearly-independent columns of A

→ can find unique vector $x_B \in \mathbb{R}^m$ s.t. $Bx_B = b$:

$$\boxed{x_B = B^{-1} b}$$

Properties of Basic Solutions

11/15/24

Lecture 19 + Lecture 20

Basic Solutions to LPs (cont.)

Claim: Let B, x_B as defined previously → can convert x_B into a soln. for $Ax=b$

[Ex.: $A=[B\ D]$ ⟶ the vector $x=[x_B\ 0]^T \in \mathbb{R}^n$ solves $Ax=b$]

→ We call x_B a basic solution with respect to basis B

- If x_B has zero entries, call it a degenerate basic soln. (not mutually exclusive!)
- If $x_B \geq 0$, call it a basic feasible soln.

Properties of Basic Solutions

Call a vector x yielding $\min_x c^T x$ under the constraints an "optimal feasible soln."

+ if x is also a basic soln., call x an "optimal basic feasible soln."

> **Theorem (Fundamental Theorem of Linear Programming)**
>
> Given an LP in standard form:
>
> (i) If ∃ a feasible solution, then ∃ a basic feasible solution
>
> (ii) If ∃ an optimal feasible solution, then ∃ an optimal basic feasible solution

Rmk: In theory, only need to check basic solutions (finitely many) to find an optimal soln. (if one exists)

⟶ in practice, have $\binom{n}{m}$ - many possible bases to check (too inefficient to brute-force)

> **Thm.:** Let $\Omega \in \mathbb{R}^n$ be the set of all feasible solutions (i.e. $x \in \mathbb{R}^n$ s.t. $Ax=b$). Then the following are equivalent:
>
> (i) x is an extreme point of Ω
>
> (ii) x is a basic feasible solution

(only need to check extreme points to find an optimal basic feas. soln (if one exists))

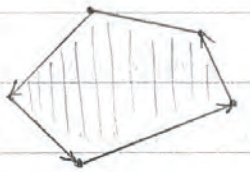
The Simplex Algorithm

11/18/24
Lecture 20 (cont.)

From previous theorems: only need to check extreme points (i.e vertices) of constraint set for an LP
→ need a systematic way to check vertices

The Simplex Algorithm (Overview)

Principle: Want to move from one vertex (feasible solution) to the next until an optimal feasible solution is found



Outline: (i) Start with basic feasible soln, $x = [x_1, \dots, x_m, 0, \dots, 0]^T$, $x_i \geq 0$ for $i = 1, \dots, m$ satisfying $x_1 a_1 + \dots + x_m a_m = b$ (where $A = [a_1 \dots a_n] \in \mathbb{R}^{m \times n}$, $m \leq n$)

(ii) At each step, replace one of the basis columns a_i for another column a_j not currently in the basis to obtain a new basic feasible solution
- Want to do in such a way that new basic soln. is also feasible
- For simplicity assume all basic solns are non-degenerate

Changing Basis Columns

Setup: Given basic feasible soln $x = [y_{10}, \dots, y_{m0}, 0, \dots, 0]^T$, want to move a new column a_q $[q > m]$ into the basis (i.e. replace some existing column a_i by a_q).

Know: $\text{rank}(A) = m \Rightarrow$ can find $y_{1q}, \dots, y_{mq}$ unique s.t.:

$$a_q = y_{10} a_1 + \dots + y_{mq} a_m \Rightarrow \boxed{\forall \varepsilon \geq 0 : (y_{10} - \varepsilon y_{1q}) a_1 + \dots (y_{m0} - \varepsilon y_{mq}) a_m + \varepsilon a_q = b}$$

Pick: $\boxed{\varepsilon = \min_i \left\{ \frac{y_{i0}}{y_{iq}} : y_{iq} > 0 \right\}}$, $p = \text{argmin}_i \left\{ y_{i0}/y_{iq} : y_{iq} > 0 \right\}$

(*) a_q "enters" basis; a_p "leaves" basis

→ Obtain a new basic feasible solution w.r.t. basis $a_1, \dots, a_{p-1}, a_{p+1}, \dots, a_m, a_q$
- If min over i achieved by more than one index i, new solution is degenerate
- If no $\lambda_i > 0$, ε arbitrarily large $\Rightarrow \Omega$ is unbounded

The Simplex Algorithm (cont.)

11/18/24

Lecture 20 The Simplex Algorithm (cont.)

+Lecture 21 Still need to determine: (i) How to pick a column to enter the basis

(ii) How to know when we have reached an optimal solution

Choosing a Basis Column

Can look at the cost before/after a_q enters the basis:

$$z_0 = c^T x = c_1 y_{10} + \dots + c_m y_{m0} \qquad z_q := c^T y_q$$

$$\rightarrow z_1 = c_1(y_{10} - \varepsilon y_{1q}) + \dots + c_m(y_{m0} - \varepsilon y_{mq}) = z_0 + (c_q - z_q)\varepsilon$$

Notice: (i) If $c_q - z_q < 0$, the new feasible soln will have a smaller $c^T x$

(ii) If $c_j - z_j \geq 0 \ \forall j = m+1, \dots, n$, then our current solution is optimal

Define reduced cost coefficients for each $i = m+1, \dots, n$: $\boxed{r_i = c_i - z_i}$

→ **Theorem:** A basic feasible solution is optimal if & only if the corresponding reduced cost coefficients are all non-negative.

(*) Notice: Previously, wrote $x = [y_{10}, \dots, y_{m0}, 0, \dots, 0]^T \in \mathbb{R}^n$

→ This is related to writing x in "augmented canonical form": $\underbrace{[I_m \ \ Y_{m,n-m}]}_{\text{becomes } y_0}\ x = y_0 = \underbrace{\begin{pmatrix} y_{10} \\ \vdots \\ y_{m0} \end{pmatrix}^T}_{\text{sent to } 0}$

Observation: $Y_{m,n-m}$ contains all y_{iq} needed for a_q to enter basis

+ after changing basis, new matrix is also in ACF → can use in next step

[Use this to notate the algorithm]

The Simplex Algorithm (cont)

11/20/24

Lecture 21

The Simplex Algorithm

1. Convert the system $Ax = b$ into ACF, and set x to be an initial feasible soln.
2. Calculate the reduced cost coefficients r_j for all non-basis variables $[i = m+1, \dots, n]$
3. If all r_j's are ≥ 0, stop $\longrightarrow$ solution is optimal

   Else, select q s.t. $r_q < 0$
4. If no y_{iq} is > 0, stop $\longrightarrow$ problem is unbounded

   Else, compute $p = \text{argmin}\{y_{i0}/y_{iq} : y_{iq} > 0\}$ ← a_q enters basis + a_p leaves basis
5. Update the matrix into new ACF by "pivoting" about $(p, q)^{th}$ element
6. Return to step 2

(*) Ex: $\min -2x_1 - 5x_2 + 0x_3 + 0x_4 + 0x_5$ (1)

s.t. $x_1 + x_3 = 4$

$x_2 + x_4 = 6$

$x_1 + x_2 + x_5 = 8$

$x_1, x_2, x_3, x_4, x_5 \geq 0$

$\longrightarrow x_{initial} = [0, 0, 4, 6, 8]^T$

basis: $B = [a_3\ a_4\ a_5]$

$c_B^T x_0 = c_1(0) + c_2(0) + c_3(4) + c_4(6) + c_5(8) = 0$

$Y_{m, n-m}$ | I_m (basis cols.)

(2) Compute r_j's: $r_1 = c_1 - z_1 = c_1 - (c_3 y_{11} + c_4 y_{21} + c_5 y_{31}) = -2$

$r_2 = c_2 - z_2 = c_2 - (c_3 y_{12} + c_4 y_{22} + c_5 y_{32}) = -5$

(3) r_j's < 0 → pick most negative r_j (common practice) → $j = 2$

(4) Compute $p = \text{argmin}\{y_{20}/y_{22}, y_{30}/y_{32}\} = \text{argmin}\{6, 8\} = 2$ $[i = 6]$

(*) $p = \text{argmin}\{y_{i0}/y_{i2} : y_{i2} > 0\}$ ($y_{i2} = 0$; $y_{22}, y_{32} > 0$)

(5) Bring a_2 into basis: $a_2 = a_4 + a_5$ ↘

$$y'_{ij} = \begin{cases} y_{ij} = y_{ij} - \frac{y_{2j}}{y_{22}} y_{i2} & i \neq 2 \\ y_{2j} = \frac{y_{2j}}{y_{22}} & i = 2 \end{cases}$$

→ New matrix (ACF):

a_1	a_2	a_3	a_4	a_5	b
1	0	1	0	0	4
0	1	0	1	0	6
1	0	0	-1	1	2

basis cols.

→ New soln: $x = [0, 6, 4, 0, 2]^T$ $(c^T x = -30)$

Intro to Duality

11/22/24

Lecture 22

(*) Two-Phase Simplex Method

So far, have assumed we have a basic feasible soln. to start ← $\binom{n}{m}$ - many possible bases

→ Not generally true for random choice of basis [that soln. is feasible]

For some LPs, can have obvious initial x

(*) Ex: LP $Ax \leq b$ → (w/ slack vars): $[A \; I_m]\begin{bmatrix} x \\ y \end{bmatrix} = b$, $\begin{bmatrix} x \\ y \end{bmatrix} \geq 0$ → pick $[0, b]^T = x_0$

In general, given standard form LP ($\min c^T x$ s.t. $Ax = b$, $x \geq 0$)

→ Can find a basic feasible soln. for LP by solving associated "artificial problem":

$$\begin{aligned} \min \; & y_1 + \dots + y_m \\ \text{s.t. } & [A \; I_m]\begin{bmatrix} x \\ y \end{bmatrix} = b \\ & \begin{bmatrix} x \\ y \end{bmatrix} \geq 0 \end{aligned}$$

← has initial basic feas. soln. $[x \; y]^T = [0 \; b]^T$

→ can use simplex alg. to solve (phase 1)

[$y_1, \dots y_m$ called "artificial variables"]

(*) Prop: The original LP has a basic feasible soln. iff the associated artificial problem has an optimal feasible soln. with $y_1 + \dots + y_m = 0$.

Duality (Symmetric Form)

Given an LP (called the primal problem), can construct its dual problem as follows:

Primal	Dual
minimize $c^T x$	maximize $\lambda^T b$
s.t. $Ax \geq b$	s.t. $\lambda^T A \leq c^T$
$x \geq 0$	$\lambda \geq 0$

where $\lambda \in \mathbb{R}^m$ is referred to as the dual vector.

Intro to Duality (cont.)

11/22/24

Lecture 22 (cont.)

Intro to Duality

Duality: central topic in many parts of optimization
- Can use dual problem to help solve original/primal problem

2 forms of duality: symmetric and asymmetric
- Symmetric form (see prev. page): Dual of the dual is the primal!
- Asymmetric form: No such relationship

Duality (Asymmetric Form)

(Primal)	(Dual)
minimize $c^T x$ s.t. $Ax = b$ $x \geq 0$	maximize $\lambda^T b$ s.t. $\lambda^T A \leq c^T$

Properties of Dual Problems

Lemma (Weak Duality): Let $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$ be feasible solutions to the primal and dual LP problems, respectively (in either symmetric or asymmetric form). Then:

$$\boxed{c^T x \geq \lambda^T b}$$

(*) Proof (Asymmetric case)

Since x and λ are both feasible, have that $Ax = b$, $x \geq 0$ [primal] and $\lambda^T A \leq c^T$ [dual].

→ Multiplying both sides of dual inequality by x:

$$\underbrace{\lambda^T A x}_{b} \leq c^T x \xrightarrow{Ax = b} \lambda^T b \leq c^T x$$

The LP Duality Theorem

11/22/24

Lecture 22 & Lecture 23

Per weak duality - any feasible soln. to either the primal/dual, gives a bound to the (optimal) cost of the other problem

- If one problem has unbounded cost, then the other prob. has no feasible soln.
- Notice: If equality holds for some $x \in \mathbb{R}^n$ & $\lambda \in \mathbb{R}^m$, then these must be optimal feasible solutions [+ can prove converse as well]

Theorem (Duality Theorem / Strong Duality)

If the primal LP problem (in either symmetric or asymmetric form) has an optimal solution, then so does the dual problem, and the optimal values for the objective functions are equal for both problems.

(*) Note: Primal/dual unbounded $\Rightarrow$ other prob. has no feasible soln.; but converse not strictly true!
Primal/dual has no feasible soln. $\Rightarrow$ other prob. may or may not have a feasible soln.
(though if it does have a feasible soln, then it must be unbounded)

Theorem (Complementary Slackness)

A pair of feasible solutions $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$ to a pair of problems is optimal iff:

$$(1)\ (c^T - \lambda^T A)x = 0 \quad \text{and} \quad (2)\ \lambda^T(Ax - b) = 0$$

Intro to Nonlinear Optimization

11/25/24

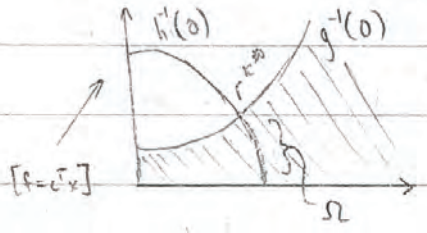
Lecture 23 (cont.)

Nonlinear Optimization with Equality Constraints

Nonlinear optimization: consider problems of the form (for $x \in \mathbb{R}^n$)

$$\boxed{\begin{aligned} &\text{minimize } f(x) \text{ subject to } h(x) = 0 \\ &\qquad\qquad\qquad\quad g(x) \leq 0 \end{aligned}} \quad \leftarrow f: \mathbb{R}^n \to \mathbb{R},\ g: \mathbb{R}^n \to \mathbb{R}^m,\ h: \mathbb{R}^n \to \mathbb{R}^p \text{ (e.g.)}$$

For 2D problems ($n=2$), can solve graphically [similar to LPs]:



$\longrightarrow$ x^* via inspection (for simple f, g, h)

Can consider the above problem with only equality constraints [i.e. w/o g: $\min f(x)$ s.t. $h(x) = 0$]

↳ **Def:** A point $x^* \in \mathbb{R}^n$ satisfying the constraints $h_1(x^*) = \dots = h_m(x^*) = 0$ is said to be **regular** if the gradients $\nabla h_i(x^*)$ are linearly independent.

(*) **Alternative defn:** x^* regular $\iff$ Dh differential of h has $\operatorname{rank}(Dh(x^*)) = m$

Lagrange Conditions

Lagrange conditions - "alternate 1st-order optimality conditions"

- Simple case: $h: \mathbb{R}^2 \to \mathbb{R}$

$\longrightarrow$ Can write $h(x) = 0$ as a curve $x(t)$ [$h(x(t)) = 0\ \forall t$]; then:

(i) $0 = \frac{d}{dt}(h \circ x)(t) = \nabla h(x(t))^T \dot{x}(t) \Rightarrow \nabla h(x(t)) \perp \dot{x}(t)\ \forall t$

(ii) Via FONC: $\frac{d}{dt}(f \circ x)(t^*) = 0$ for $t^* = \operatorname{argmin}_t f(x(t))$

$\Rightarrow \nabla f(x(t^*)) \perp \dot{x}(t^*)$

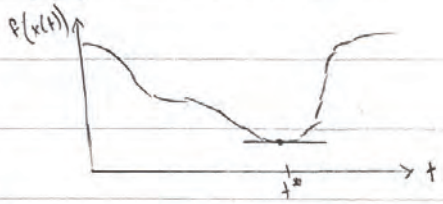
$\Rightarrow \underline{\nabla f(x(t^*)) \parallel \nabla h(x(t^*))}$

Lagrange's Theorem

11/25/24

Lecture 23 + Lecture 24

Lagrange Conditions (cont.)



Previously: found that $\nabla f(x(t^*)) \parallel \nabla h(x(t^*))$

$\Rightarrow \underline{\exists\ \lambda \in \mathbb{R} \text{ s.t. } \nabla f(x(t^*)) = \lambda \nabla h(x(t^*))}$

Thm. (1st-Order Optimality Conditions) ($h: \mathbb{R}^n \to \mathbb{R}$)

If x^* is a local min. s.t. $h(x^*) = 0$ and $\nabla h(x^*) \neq 0$, then $\exists\ \lambda^* \in \mathbb{R}$ such that:

$$\boxed{\nabla f(x^*) + \lambda^* \nabla h(x^*) = 0}$$ [Necessary, but ~~not~~ sufficient]

↓ (generalization)

Theorem (Lagrange's Theorem)

Let $x^* \in \mathbb{R}^n$ be a local min/max of $f: \mathbb{R}^n \to \mathbb{R}$ s.t. $h(x) = 0$ for $h: \mathbb{R}^n \to \mathbb{R}^m$. Assume x^* is regular; then $\exists\ \lambda^* \in \mathbb{R}^m$ such that:

$$\boxed{\nabla f(x^*) + Dh(x^*)^T \lambda^* = 0}$$

(*) Note: Assumption of x^* regular is nontrivial

The Lagrangian

Def: The Lagrangian function is the function $\ell: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by:

$$\boxed{\ell(x, \lambda) = f(x) + h(x)^T \lambda}$$

→ Can represent the Lagrangian condition by: ← FONC of the Lagrangian

$$\boxed{\nabla \ell(x^*, \lambda^*) = 0}$$ [for some $x^* \in \mathbb{R}^n$, $\lambda^* \in \mathbb{R}^m$]

The Lagrangian

11/27/24

Lecture 24 (cont.)

The Lagrangian (cont.)

The Lagrangian condition is the FONC of the Lagrangian:

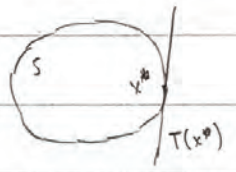
$$\nabla \ell(x,\lambda) = \begin{bmatrix} \nabla_x \ell(x,\lambda) \\ \nabla_\lambda \ell(x,\lambda) \end{bmatrix} = \begin{bmatrix} \nabla f(x) + Dh(x)^T \lambda \\ h(x) \end{bmatrix} \stackrel{?}{=} 0$$

→ can often find minimizers to the original problem via solving the Lagrangian condition $\nabla \ell(x,\lambda) = 0$.

Second-Order Conditions

Def: Let S be a surface $S := \{x \in \mathbb{R}^n : h(x) = 0\}$, and let $x^* \in S$. Then the tangent space at x^* on S is defined as:

$$T(x^*) = \{y \in \mathbb{R}^n : Dh(x^*)y = 0\}$$



Theorem (2nd-Order Necessary Conditions - Equality Constraints)

Let $x^* \in \mathbb{R}^n$ be a local minimizer of $f: \mathbb{R}^n \to \mathbb{R}$ s.t. $h(x) = 0$, $h: \mathbb{R}^n \to \mathbb{R}^m$ $[m \leq n]$ and $f, h \in C^2(\mathbb{R}^n)$. Assume x^* is regular. Then $\exists\ \lambda^* \in \mathbb{R}^m$ such that:

(1) $\nabla f(x^*) + Dh(x^*)^T \lambda^* = 0$

(2) $\forall\ y \in T(x^*)$, we have that $y^T D^2 \ell(x^*, \lambda^*) y \geq 0$ [P.S.D. on $T(x^*)$]

Theorem (2nd-Order Sufficient Conditions - Equality Constraints)

Suppose $f, h \in C^2(\mathbb{R}^n)$, and let $x^* \in \mathbb{R}^n$, $\lambda^* \in \mathbb{R}^m$ be a pair of points such that:

(1) $\nabla f(x^*) + Dh(x^*)^T \lambda^* = 0$

(2) $\forall\ y \in T(x^*)$ $[y \neq 0]$, we have that $y^T D^2 \ell(x^*, \lambda^*) y > 0$. [P.D. on $T(x^*)$]

→ Then x^* is a strict local minimizer of $f(t)$ s.t. $h(x) = 0$.

The KKT Conditions

11/27/24

Lecture 24 (cont.)

Nonlinear Optimization with Inequality Constraints

Consider more general problems:

$$\text{minimize } f(x) \text{ s.t. } h(x) = 0,\ g(x) \leq 0$$

Def: An inequality constraint $g_j(x) \leq 0$ is called active at point $x^* \in \Omega$ if $g_j(x^*) = 0$, and inactive if $g_j(x^*) < 0$.

- By convention, say that a constraint $h_i(x) = 0$ is always active

Def: Let $x^* \in \Omega$ satisfy $h(x^*) = 0$ and $g(x^*) \leq 0$. Let $J(x^*)$ denote the index set of active inequality constraints:

$$J(x) := \{ j : g_j(x^*) = 0 \}$$

Then we say that x^* is a regular point if:

$$\{ \nabla h_i(x^*) : i = 1, \dots, m \} \cup \{ \nabla g_j(x^*) : j \in J(x^*) \} \text{ are linearly independent}$$

Theorem (Karush-Kuhn-Tucker)

Let $f, h, g \in C^1(\mathbb{R}^n)$. Let x^* be a regular point and a local minimizer of f s.t. $h(x) = 0$, $g(x) \leq 0$. Then $\exists\ \lambda^* \in \mathbb{R}^m$, $\mu^* \in \mathbb{R}^p$ such that:

(1) $\mu^* \geq 0$

(2) $\nabla f(x^*) + Dh(x^*)^T \lambda^* + Dg(x^*)^T \mu^* = 0$

(3) $(\mu^*)^T g(x^*) = 0$

↳ Call these the KKT conditions

- λ^* called the Lagrange multiplier; μ^* called the KKT multiplier

The KKT Conditions (cont.)

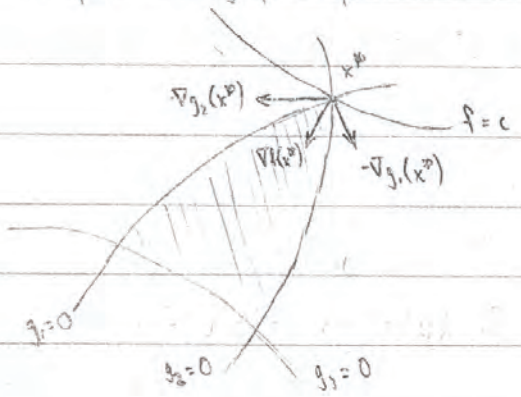
12/2/24

Lecture 25

Notice: KKT conditions stipulate (i) $\mu^* \geq 0$, and (ii) $(\mu^*)^T g(x^*) = 0$ ← active ⇒ may or may not be 0

→ $\forall\ g_j(x^*) < 0$, have that $\mu_j^* = 0$ [KKT multipliers corresponding to inactive constraints are always 0]

Can interpret KKT graphically:



↱ by KKT:

- Constraint g_3 inactive ⇒ $\mu_3^* = 0$

→ Have: $\nabla f(x^*) + \mu_1^* \nabla g_1(x^*) + \mu_2^* \nabla g_2(x^*) = 0$

$$\Updownarrow$$

$$\nabla f(x^*) = -\mu_1^* \nabla g_1(x^*) - \mu_2^* \nabla g_2(x^*)$$

→ **Consequence:** By KKT, $\nabla f(x^*)$ must be a linear comb. of $-\nabla g_1(x^*), -\nabla g_2(x^*)$ w/ non-negative coeffs.

Second-Order Conditions

Can define a mapping $\ell : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ [similar to Lagrangian]:

$$\boxed{\ell(x, \lambda, \mu) = f(x) + h(x)^T \lambda + g(x)^T \mu}$$

→ To derive 2nd-order conditions, can look at its Hessian:

$$\boxed{D^2 \ell(x, \lambda, \mu) = D^2 f(x) + \sum_{i=1}^{m} \lambda_i D^2 h_i(x) + \sum_{j=1}^{p} \mu_j D^2 g_j(x)}$$

+ Define **tangent space generated by the active constraints:**

$$\boxed{\{ y \in \mathbb{R}^n : Dh(x^*) y = 0,\ Dg_j(x^*) y = 0\ \forall\ j \in J(x^*) \}}$$

The KKT Conditions (cont.)

12/2/24

Lecture 25

Theorem (2nd-Order Necessary Conditions)

Let x^* be a local min of $f: \mathbb{R}^n \to \mathbb{R}$ s.t. $h(x) = 0$, $g(x) \leq 0$ for $h: \mathbb{R}^n \to \mathbb{R}^m$, $g: \mathbb{R}^n \to \mathbb{R}^p$, and $f, g, h \in C^2(\mathbb{R})$. Assume x^* is regular. Then $\exists$ $\lambda^* \in \mathbb{R}^n$, $\mu^* \in \mathbb{R}^p$ such that:

(1) $\mu^* \geq 0$, $\nabla f(x^*) + Dh(x^*)^T \lambda^* + Dg(x^*)^T \mu^* = 0$, $(\mu^*)^T (g(x^*)) = 0$ [KKT]

(2) $\forall\, y \in T(x^*)$, we have that $y^T D^2 \mathcal{L}(x^*, \lambda^*, \mu^*) y \geq 0$.

Theorem (2nd-Order Sufficient Conditions)

Let $f, g, h \in C^2(\mathbb{R}^n)$ and let $x^* \in \mathbb{R}^n$ feasible, $\lambda^* \in \mathbb{R}^m$, and $\mu^* \in \mathbb{R}^p$ such that:

(1) $\mu^* \geq 0$, $\nabla f(x^*) + Dh(x^*)^T \lambda^* + Dg(x^*)^T \mu^* = 0$, $(\mu^*)^T (g(x^*)) = 0$ [KKT]

(2) $\forall\, y \in T(x^*)$, we have that $y^T D^2 \mathcal{L}(x^*, \lambda^*, \mu^*) y > 0$.

$\Rightarrow$ Then x^* is a strict local minimizer of f s.t. $h(x) = 0$, $g(x) \leq 0$.

(*) Note: Can derive similar results for maximizers

Projection Methods

12/04/24

Lecture 26

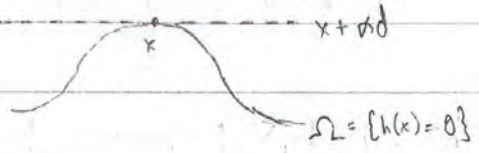
Algorithms for Constraint Optimization

Want to solve general problems: $\min f(x)$ s.t. $x \in \Omega \subseteq \mathbb{R}^n$

(i) Projection Methods

Previously saw: for some constraints $h(x) = 0$, may not have a feasible direction (per se)

[i.e. for some $x \in \Omega$, may not exist $d \neq 0$ s.t. $x + \alpha d \in \Omega \ \forall \ \alpha \in [0,1]$]



→ One way to solve: via projecting $x + \alpha d$ onto Ω

$$x^{k+1} := \Pi\left(x^k + \alpha_k d^k\right)$$

where $\Pi(x) = \underset{z \in \Omega}{\operatorname{argmin}} \|z - x\|^2$

↳ May work well in some cases (esp. where $\Pi(x)$ is known/has closed-form solution), but often is intractable (finding $\Pi(x)$ is its own optimization problem, potentially as difficult [or harder] than the original problem itself)

(*) Ex. [Tractable]: Ω is of linear variety $\Omega = \{Ax = b\}$, $A \in \mathbb{R}^{m \times n}$, $\operatorname{rank}(A) = m < n$

→ $\Pi(x) = Px$, where $P = I_n - A^T(AA^T)^{-1}A$ [closed-form]

→ Update step' (assuming x^k feasible): $x^{k+1} = x^k - \alpha_k P \nabla f(x^k)$ → $P\nabla f(x^*) = 0$ for x^* satisfying Lagrange conditions

Rmk: A lot of theory for unconstrained gradient methods can be generalized to the constrained case.

Lagrangian Algorithms

12/04/24

Lecture 26 (cont.)

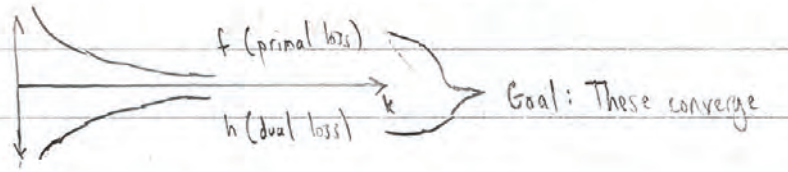
(ii) Lagrangian Algorithms

For problems $[\min f(x) \text{ s.t. } h(x)=0;\ f:\mathbb{R}^n\to\mathbb{R},\ h:\mathbb{R}^n\to\mathbb{R}^m]$, can recall the Lagrangian

$\ell(x,\lambda) = f(x) + \lambda h(x) \quad [\lambda\in\mathbb{R}^m]$

→ Lagrangian algorithms use standard gradient algorithms on $\ell(x,\lambda)$ to (i) minimize $\ell(x,\lambda)$ w.r.t. x, and (ii) maximize $\ell(x,\lambda)$ w.r.t. λ

$$\nabla\ell(x,\lambda) = \begin{bmatrix}\nabla_x \ell \\ \nabla_\lambda \ell\end{bmatrix} \begin{cases} \nearrow\ x^{k+1} = x^k - \alpha_k \nabla_x \ell(x,\lambda) = x^k - \alpha_k \nabla f(x^k) + Dh(x^k)^T \lambda^k \\ \searrow\ \lambda^{k+1} = \lambda^k + \beta_k \nabla_\lambda \ell(x,\lambda) = \lambda^k + \beta_k h(x^k) \end{cases}$$



Rmks: (i) The fixed points of the algorithm are the points satisfying Lagrange conditions (& vice versa)

(ii) Lagrange algorithm (above) converges locally with worst-case linear rate

- For certain kinds of f, h: may converge globally

Can consider case w/ inequality constraints: $[\min f(x) \text{ s.t. } g(x)\le 0;\ g:\mathbb{R}^n\to\mathbb{R}^p]$; $\ell(x,\mu) = f(x) + g(x)^T\mu$

→ Lagrange algorithm (via KKT):

(i) $x^{k+1} = x^k - \alpha_k \nabla f(x^k) + Dg(x^k)^T \mu$

(ii) $\mu^{k+1} = \max\{0,\ \mu^k + \beta_k g(x^k)\}$ ← notation: $[\cdot]_+ = \max(0,\cdot)$

↳ Due to $\mu\ge 0$ constraint

Rmks: (i) μ^{k+1} update step is given by $\pi_{\mathbb{R}^p_+}(\mu^k + \beta_k g(x^k)) = \underset{z\in\mathbb{R}^p_+}{\operatorname{argmin}} \|z - (\mu^k + \beta_k g(x^k))\|^2$

(ii) Fixed points of above alg. correspond to points satisfying KKT conditions (& vice versa)

(iii) Similar to before: local convergence w/ worst-case linear rate; global for some f, g

Penalty Methods

12/24/24
Lecture 26
(cont.)
FINAL

(iii) Penalty Methods

For general problems [$\min f(x)$ s.t. $x \in \Omega$], try to solve:

$$\boxed{\min f(x) + \gamma P(x)} \quad \text{[unconstrained]}$$

where $\gamma > 0$ and $P: \mathbb{R}^n \to \mathbb{R}$ is a penalty satisfying:

(i) $P(x)$ is continuous

(ii) $P(x) > 0 \ \forall \ x \notin \Omega$

(iii) $P(x) = 0 \iff x \in \Omega$

(*) Ex. (Penalty): $\Omega := \{x : g_j(x) \leq 0, \ j = 1, \dots, p\}$

$$\to P(x) = \sum_{j=0}^{p} \max\left(0, g_j(x)\right)$$ ← (*) Notation: $g_j^+(x)$

Note: Can write equality constraints $\{h(x) = 0\}$ as, equivalently, $\{\|h(x)\|^2 \leq 0\}$

(*) Penalty methods: can increase γ to push solution closer to Ω & repeat for multiple γ to obtain an approximation of x^*

- Penalty method solution $x^*_{\text{penalty}} \to x^*$ as $\gamma \to \infty$