Aug 24st. 2021

§ Fundamentals of optimization

An optimization problem can be written as

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{aligned} c_i(x) &= 0 \quad i \in \mathcal{E} \\ c_i(x) &\geq 0 \quad i \in \mathcal{I} \end{aligned}$$

f: objective function

x: vector of variables, also called unknowns or parameters

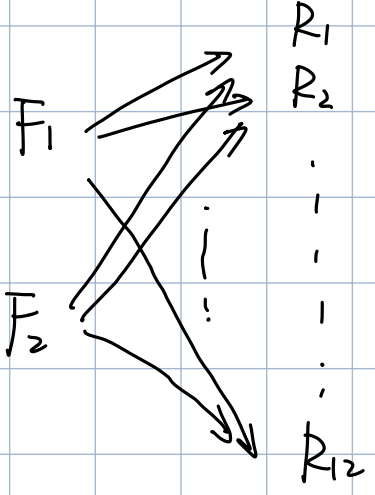
c_i: constraint function

$\mathcal{I}, \mathcal{E}$: sets of indices for inequality and equality constraints, respectively

Ex. A chemical company has two factories F_1 and F_2 and a dozen outlets $R_1, R_2, \cdots, R_{12}$. Each factory F_i can produce a_i tons of a certain chemical product each week, a_i is called the capacity of the plant. Each retail outlet R_j has a known weekly demand of b_j tons of the product. The cost of shipping one ton of the product from factory F_i to retail outlet R_j is c_{ij}.

The problem is to determine how much of the product to ship from each factory to each outlet so as to satisfy all the requirements and minimize cost.

Let $x_{ij}, i = 1, 2, j = 1, 2, \cdots, 12$ be the number of tons of product shipped from F_i to R_j



$$\min \sum_{i,j} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{j=1}^{12} x_{ij} \leq a_i, \quad i = 1, 2$$

$$\sum_{i=1}^{2} x_{ij} \geq b_j, \quad j = 1, 2, \cdots, 12$$

$$x_{ij} \geq 0, \quad i = 1, 2, \; j = 1, 2, \cdots, 12$$

This type of problem is known as a linear programming problem

$$f(x) = \sum_{ij} c_{ij} x_{ij} \qquad x = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1,12} \\ x_{21} \\ \vdots \\ x_{2,12} \end{pmatrix} \rightarrow 2+12+24$$

$$\mathcal{E} = \emptyset \qquad I = \{1, 2, \cdots, 38\}$$

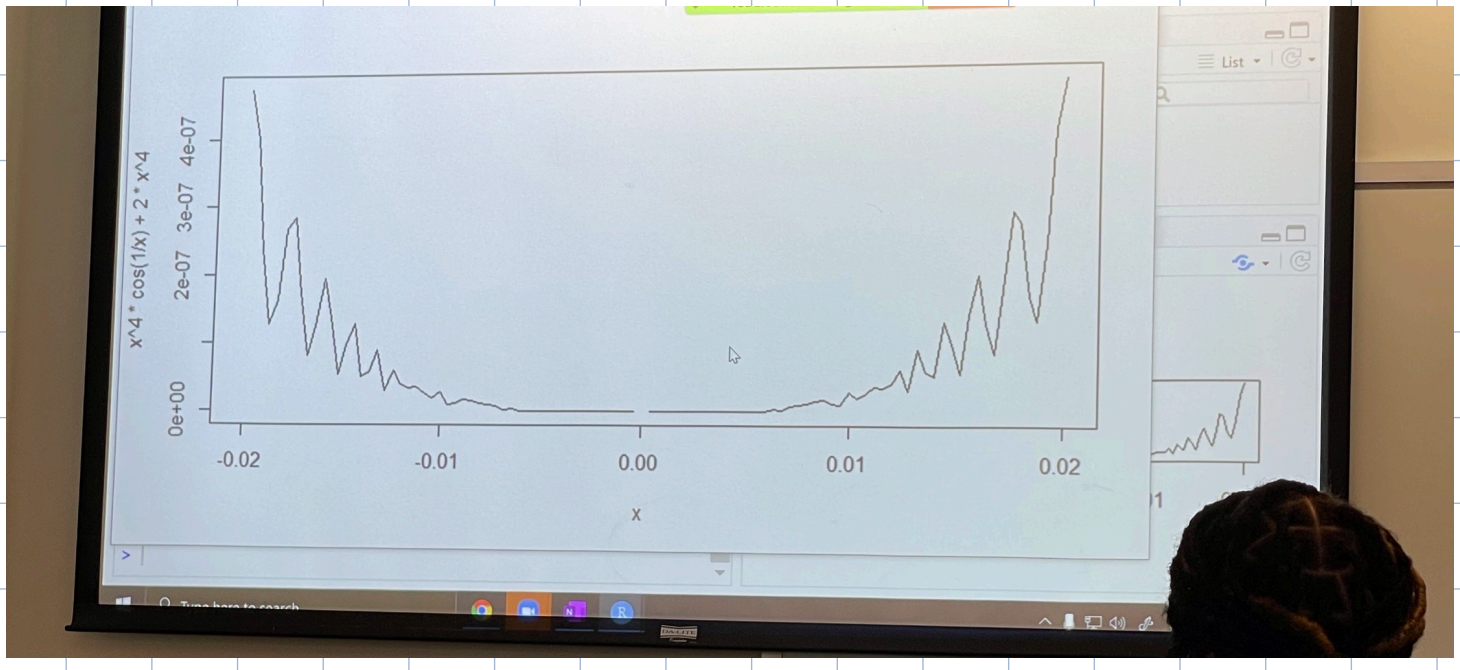
$$c(x) = \begin{pmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_{38}(x) \end{pmatrix} = \begin{pmatrix} -\sum_{j=1}^{12} x_{1j} + a_1 \\ \vdots \\ \sum_{i=1}^{2} x_{i1} - b_1 \\ \vdots \\ x_{2,12} \end{pmatrix}$$

Def. A point x^* is a global minimizer if $f(x^*) \leq f(x)$ for all x

Def. A point x^* is a local minimizer if there is a neighborhood N of x^* such that $f(x^*) \leq f(x)$ for all $x \in N$ (week local minimizer)

Def. A point x^* is a local minimizer if there is a neighborhood N of x^* such that $f(x^*) < f(x)$ for all $x \in N$ (strong local minimizer)

Ex. $f(x) = x^4 \cos(\frac{1}{x}) + 2x^4$



global minimizers: 0 $(f(0) = 0)$

local minimizers: $\frac{1}{(2k+1)\pi}$ for $k \in \mathbb{N}$

Def: A point x^* is an isolated local minimizer if there is a neighborhood N of x^* such that x^* is the only local minimizer in N.

Def: A point x^* is a stationary point if $\nabla f(x^*) = 0$
(f should be differentiable in this case)

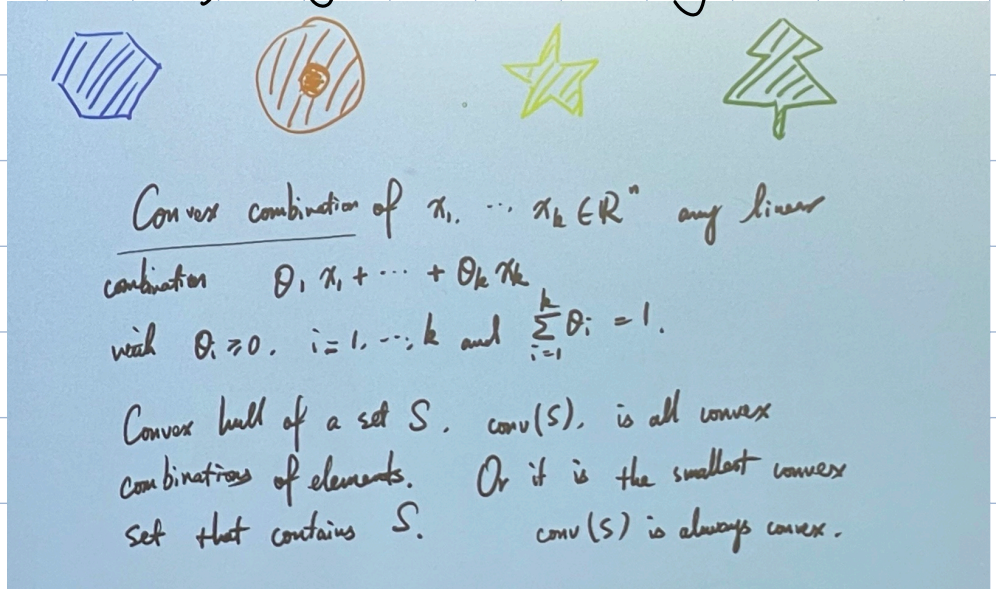
Theorem: If x^* is a local minimizer and f is continuously differentiable in an open neighborhood of x^*, then $\nabla f(x^*) = 0$
(first-order necessary condition)

Theorem: If x^* is a local minimizer of f and $\nabla^2 f$ exists and is continuous in an open neighborhood of x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f$ is positive semidefinite.
(second-order necessary condition)

Theorem: Suppose that $\nabla^2 f$ is continuous in an open neighborhood of x^* and that $\nabla f(x^*) = 0$, and $\nabla^2 f(x^*)$ is positive definite, then x^* is a strict local minimizer of f.

§ convex sets and convex functions

For any $x, y \in S$, $tx + (1-t)y \in S \longrightarrow S$ convex



Convex combination of $x_1, \cdots x_k \in \mathbb{R}^n$ any linear combination $\theta_1 x_1 + \cdots + \theta_k x_k$ with $\theta_i \geq 0$, $i = 1, \cdots, k$ and $\sum_{i=1}^{k} \theta_i = 1$.

Convex hull of a set S. conv(S), is all convex combinations of elements. Or it is the smallest convex set that contains S. conv(S) is always convex.

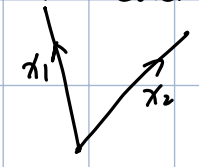
Examples of convex sets:

- Line
- Norm ball $\{x: \|x\| \le r\}$ for given norm $\|\cdot\|$ and radius r
- Hyperplane $\{x: a^T x = b\}$ for given a, b
- Halfspace $\{x: a^T x \le b\}$ for given a, b
- Affine space $\{x: Ax = b\}$ for given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$
- Polyhedron $\{x: Ax \le b\}$ for given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

  The set $\{x: Ax \le b, Cx = d\}$ is also a polyhedron

  e.g. $Cx \le d$, $-Cx \le -d$

- Probability simplex $\{w: w \ge 0, 1^T w = 1\}$
- Cone $S \subseteq \mathbb{R}^n$ such that $x \in S \Rightarrow tx \in S$ for all $t \ge 0$



non-convex cone

<u>convex cone</u>: cone that is also convex

$x_1, x_2 \in S \Rightarrow t_1 x_1 + t_2 x_2 \in S$ for all $t_1, t_2 \ge 0$

<u>Conic combination</u> of $x_1, \dots, x_k \in \mathbb{R}^n$ any linear combination $\sum_{i=1}^{k} \theta_i x_i$ with $\theta_i \ge 0$

Examples of convex cones

- Norm cone $\{(x, t): \|x\| \le t\}$

  Under the ℓ_2 norm $\|\cdot\|_2$, it is called second-order cone.
- Positive semidefinite cone $S_+^n = \{X \in S^n \; X \ge 0\}$

Operations preserving convexity

- Intersection $S_1 \cap S_2$
- Scaling and translation $S \to aS + b$
- Affine image and preimage

  if $f(x) = Ax + b$, and C is convex, then $f(C)$ is convex

if D is convex, then $f^{-1}(D)$ is convex

Convex functions $f: \mathbb{R}^n \to \mathbb{R}$ such that $\text{dom}(f) \subseteq \mathbb{R}^n$ convex, and $f(tx+(1-t)y) \le tf(x)+(1-t)f(y)$ for $0 \le t \le 1$ and all $x, y \in \text{dom}(f)$.
(rewords, function lies below the line segment joining $f(x), f(y)$)

f convex $\Leftrightarrow$ $-f$ concave

Important modifiers:

- Strictly convex $f(tx+(1-t)y) < tf(x)+(1-t)f(y)$ for all $0<t<1$, $x \neq y$
  In words, f is convex and has greater curvature than a linear func
- Strongly convex with parameter $m>0$
  $f - \frac{m}{2}\|x\|_2^2$ is convex
  In words, f is at least as convex as a quadratic func

Note: Strongly convex $\Rightarrow$ strictly convex $\Rightarrow$ convex

Examples of convex functions:

- Exponential func e^{ax} is convex for $a \in \mathbb{R}$
- Power func x^a is convex for $a \ge 1$ or $a \le 0$ (for $x \in \mathbb{R}$ / for $x > 0$)
  x^a is concave for $0 \le a \le 1$
- Logarithm func $\log x$ is concave over $x>0$
- Affine func $a^T x + b$ is convex and concave
- Quadratic func $\frac{1}{2}x^T Q x + b^T x + c$ is convex provided that $Q \ge 0$
  Least square loss $\|y - Ax\|_2^2$ is always convex (since $A^T A \ge 0$)
- Norm $\|x\|$ is convex for any norm e.g. l_p norm $\|x\|_p = \left(\sum_{i=1}^n x_i^p\right)^{1/p}$ for $p \ge 1$
  $\|x\|_\infty = \max_i |x_i|$

  Operator (Spectral) norm: $\|X\|_{op} = \sigma_1(X)$ for $X \in \mathbb{R}^{m \times n}$, $X > 0$
  Nuclear (trace) norm: $\|X\|_{tr} = \sum_{i=1}^r \sigma_i(X)$

$\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_r(X) \geq 0$ are singulars of X.

· Indicator func if S is convex, then its indicator func is

$$I_S(x) = \begin{cases} 0 & \text{if } x \in S \\ \infty & \text{if } x \notin S \end{cases} \quad \text{is convex}$$

· Max func $f(x) = \max\{x_1, \cdots, x_n\}$ is convex

Operators preserving convexity:

· Non-negative linear combination $f_1, \cdots, f_m$ convex $\Rightarrow \sum_{i=1}^{m} a_i f_i$ convex for $a_i \geq 0$

· Pointwise maximization f_s convex for any $s \in S \Rightarrow f(x) = \max_{s \in S} f_s(x)$ convex

Note that the set S can have infinite number of elements

· Partial minimization $g(x,y)$ convex, C convex $\Rightarrow f(x) = \min_{y \in C} g(x,y)$ convex

Ex. Let C be an arbitrary closed set, and consider the maximum distance to C under an arbitrary norm $\|\cdot\|$

$f(x) = \max_{y \in C} \|x - y\|$ is convex

Let C be a convex closed set, and consider the minimum distance to C under an arbitrary norm $\|\cdot\|$

$f(x) = \min_{y \in C} \|x - y\|$ is convex

C convex $\Leftrightarrow \min_{y \in C} \|x - y\|$ convex

Aug 26st 2021

· Affine composition f convex $\Rightarrow g(X) = f(AX+b)$ convex

· General composition $f = h \circ g$, where $g: \mathbb{R}^n \to \mathbb{R}$, $h: \mathbb{R} \to \mathbb{R}$

g convex
h convex and nondecreasing
$\Rightarrow f$ convex

g concave, h convex and nonincreasing $\Rightarrow f$ convex

g concave, h concave and nondecreasing $\Rightarrow f$ concave

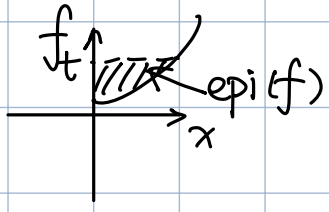
g convex, h concave and nonincreasing $\Rightarrow f$ concave

$f'(x) = h'(g)g'(x)$

$f''(x) = h''(g)(g'(x))^2 + h'(g)g''(x)$ ← used to check above

Properties of convex functions

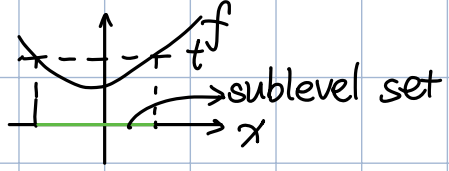
- f convex $\Leftrightarrow$ epigraph of f is a convex set



$$\text{epi}(f) = \{(x,t) \in \text{dom}(f) \times \mathbb{R} : f(x) \le t\}$$

- f convex $\Rightarrow$ sublevel sets of f are convex for all $t \in \mathbb{R}$

$$\{x \in \text{dom}(f) : f(x) \le t\}$$



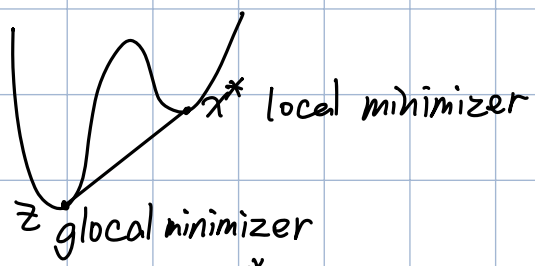
△First-order characterization

If f is differentiable, then f convex $\Leftrightarrow$ $\text{dom}(f)$ convex and

$f(y) \ge f(x) + \nabla f(x)^T(y-x)$ for all $x, y \in \text{dom}(f)$

Theorem:

f is differentiable, then any stationary point



$x = \lambda z + (1-\lambda)x^*$

$f(x) \le \lambda \underbrace{f(z)}_{\le f(x^*)} + (1-\lambda) f(x^*) \le f(x^*)$

△ Second-order characterization

If f is twice differentiable, then

f convex $\Leftrightarrow$ dom(f) convex and $\nabla^2 f(x)$ is positive definite

· Jensen's inequality

If X is a random variable supported on dom(f) and f is convex, then $f(E(X)) \le E(f(X))$

Ex. $\min_x \; x_1 e^{-(x_1^2+x_2^2)}$

$$\nabla f(x) = \begin{pmatrix} e^{-(x_1^2+x_2^2)} - 2x_1^2 e^{-(x_1^2+x_2^2)} \\ -2x_1x_2 e^{-(x_1^2+x_2^2)} \end{pmatrix} = \begin{pmatrix} (1-2x_1^2)e^{-(x_1^2+x_2^2)} \\ -2x_1x_2 e^{-(x_1^2+x_2^2)} \end{pmatrix}$$

$\nabla f(x) = 0$ at $x^* = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$ or $x^{**} = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$

$$\nabla^2 f(x) = \begin{pmatrix} (4x_1^3 - 6x_1)e^{-(x_1^2+x_2^2)} & (4x_1^2x_2 - 2x_2)e^{-(x_1^2+x_2^2)} \\ (4x_1^2x_2 - 2x_2)e^{-(x_1^2+x_2^2)} & (4x_1x_2^2 - 2x_1)e^{-(x_1^2+x_2^2)} \end{pmatrix}$$

$$\nabla^2 f(x^*) = \begin{pmatrix} -2\sqrt{2}e^{-\frac{1}{2}} & 0 \\ 0 & -\sqrt{2}e^{-\frac{1}{2}} \end{pmatrix} < 0 \Rightarrow x^* \text{ is a strict local maximizer}$$

$$\nabla^2 f(x^{**}) = \begin{pmatrix} 2\sqrt{2}e^{-\frac{1}{2}} & 0 \\ 0 & \sqrt{2}e^{-\frac{1}{2}} \end{pmatrix} > 0 \Rightarrow x^{**} \text{ is a strict local minimizer}$$

Ex. $\min_x \; (x_2 - x_1^2)^2 + x_1^5$

$$\nabla f(x) = \begin{pmatrix} 5x_1^4 - 4x_1(x_2 - x_1^2) \\ 2(x_2 - x_1^2) \end{pmatrix}$$

$\nabla f(x) = 0$ at $x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$\nabla^2 f(x) = \begin{pmatrix} 20x_1^3 + 12x_1^2 - 4x_2 & -4x_1 \\ -4x_1 & 2 \end{pmatrix}$$

$$\nabla^2 f(x^*) = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} \geq 0$$

x^* is a stationary point, neither a local minimizer nor a local maxi

Rate of convergence

Beginning at x_0, optimization algorithms generate a sequence of iterates $\{x_k\}_{k=0}^{\infty}$ that terminate at some stopping condition.

Denote the actual local minimizer as x^*

We say that the convergence is Q-linear if there is a constant $r \in (0,1)$, such that

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq r \quad \text{for all } k \text{ sufficiently large}$$

e.g $x_k = 1 + 0.5^k$

We say that the convergence is Q-superlinear if $\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$

e.g $x_k = 1 + k^{-k}$

△Line Search Methods

For a general algorithm

Initialize $x_0 (k=0)$

while stopping condition is not satisfied at x_k

(a) Find x_{k+1} such that $f(x_{k+1}) < f(x_k)$

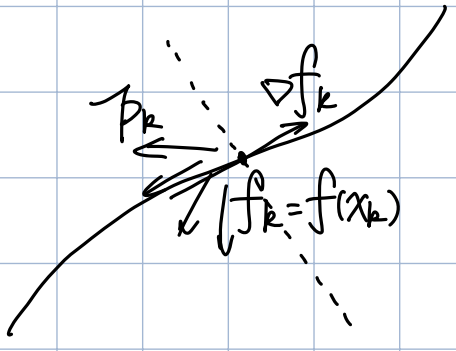
(b) $k = k+1$

end

Output $x^* = x_k$

Linear search method

$$x_{k+1} = x_k + \alpha_k p_k$$



α_k: step length

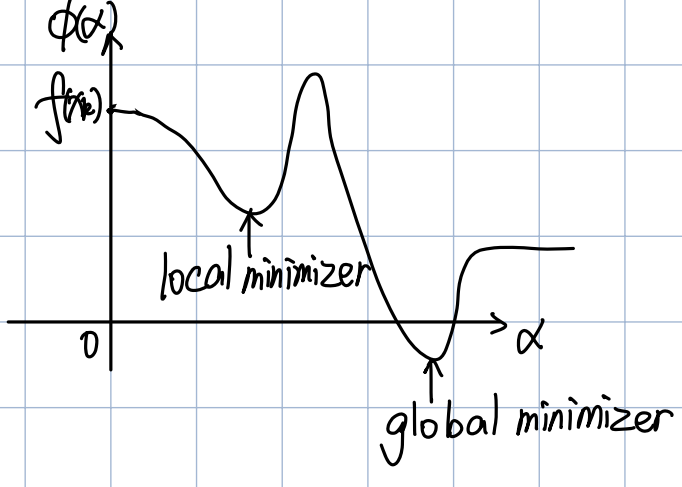
p_k: descent direction $p_k^T \nabla f_k < 0$

p_k often takes form of $p_k = -B_k^{-1} \nabla f_k$, where B_k is a symmetric and nonsingular matrix

- $p_k = -\nabla f_k$ steepest descent method
- $p_k = -\nabla^2 f(x_k)^{-1} \nabla f_k$ Newton's method

Choose step length

$$\phi(\alpha) = f(x_k + \alpha p_k) \quad \alpha > 0$$



inexact line search

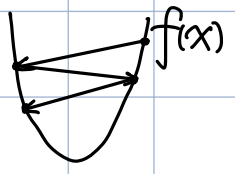
Two common issues in line search methods

(a) $\min x^2$

$x_k = (-1)^k(1+2^{-k})$

k	0	1	2	3
x_k	2	$-\frac{3}{2}$	$\frac{5}{4}$	$-\frac{9}{8}$
$f(x_k)$	4	$\frac{9}{4}$	$\frac{25}{16}$	$\frac{81}{64}$

$f(x_{k+1}) < f(x_k)$



$\{x_k\}$ does not converge

$\{f(x_k)\}$ does not converge to local or global minimum

Issue: Small decrease in function values relative to step length

Armijo condition: ensures sufficient decrease in the func value

(b) $\min x^2$

$x_k = 1+2^{-k}$

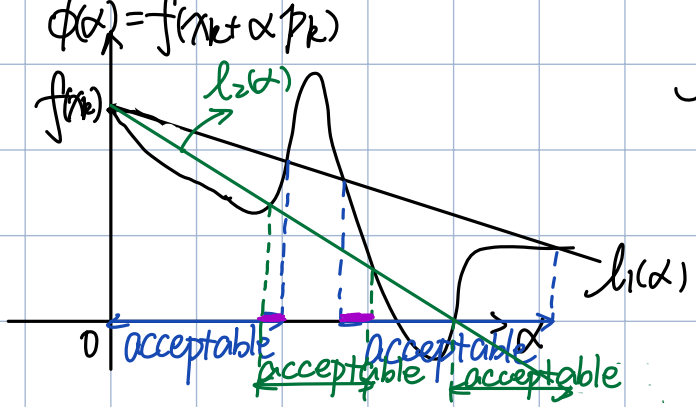
k	0	1	2	3
x_k	2	$\frac{3}{2}$	$\frac{5}{4}$	$\frac{9}{8}$
$f(x_k)$	4	$\frac{9}{4}$	$\frac{25}{16}$	$\frac{81}{64}$

$f(x_{k+1}) < f(x_k)$

$\lim_{k\to\infty} x_k = 1$, not local minimizer

Issue: Step sizes are too small relative to the initial decrease of f

Wolfe condition



$l_1(\alpha) = f(x_k) + c_1\alpha \underbrace{\nabla f_k^T p_k}_{<0}$

$c_1 \in (0,1)$

In practice, c_1 is chosen to be quite small, say $c_1 = 10^{-4}$

$f(x_k+\alpha p_k) \le l(\alpha) = f(x_k)+c_1\alpha\nabla f_k^T p_k$ ⟶ Armijo condition

Armijo-Goldstein condition ensures that the step lengths are not too small

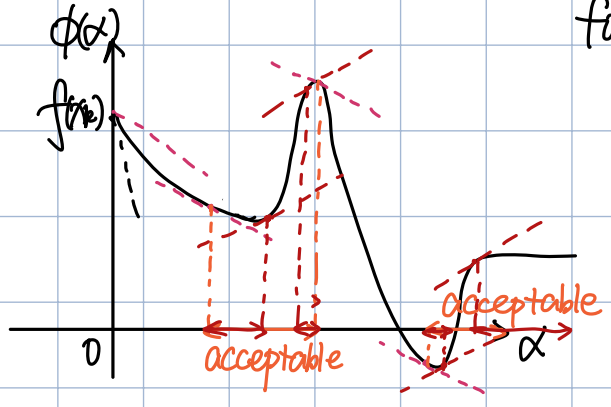
$$f(x_k)+(1-c)\alpha_k\nabla f_k^T p_k \le \underbrace{f(x_k+\alpha_k p_k)}_{=l_2(\alpha)} \le f(x_k)+c\alpha_k\nabla f_k^T p_k$$

Goldstein's condition — Armijo's condition

Disadvantage: it may exclude all minimizers of ϕ.
(see purple part of the graph)

Wolfe condition ensures sufficient rate of decrease of function value in the given direction

curvature condition $\nabla f(x_k+\alpha_k p_k)^T p_k \ge c_2\nabla f_k^T p_k$
for some constant $c_2 \in (c_1, 1)$



Wolfe conditions

$$f(x_k+\alpha_k p_k) \le f(x_k)+c_1\alpha_k\nabla f_k^T p_k$$
$$\nabla f(x_k+\alpha p_k)^T p_k \ge c_2\nabla f_k^T p_k$$
$$0 < c_1 < c_2 < 1$$

Strong Wolfe conditions

$$f(x_k+\alpha_k p_k) \le f(x_k)+c_1\alpha_k\nabla f_k^T p_k$$
$$|\nabla f(x_k+\alpha p_k)^T p_k| \le c_2|\nabla f_k^T p_k|$$
$$0 < c_1 < c_2 < 1$$

We no longer allow the derivative $\phi'(\alpha_k)$ to be too positive.

Hence we exclude points that are far away from local minimizer or stationary point

Existence of such step length

Lemma: Suppose that $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Let p_k be a descent direction at x_k, and assume that f is bounded below along the direction $\{x_k + \alpha p_k, \alpha > 0\}$. Then if $0 < c_1 < c_2 < 1$, there exist intervals of steps satisfying the Wolfe conditions and the strong Wolfe conditions.

Proof: Since $0 < c_1 < 1$, the line $l_1(\alpha) = f(x_k) + \alpha c_1 \nabla f_k^T p_k$ is unbounded below, and must intersect the graph of $\phi(\alpha)$ at least once.

Let $\alpha' > 0$ be the smallest intersecting value of α.

$$f(x_k + \alpha' p_k) = f(x_k) + \alpha' c_1 \nabla f_k^T p_k$$

Then Armijo condition clearly holds for all step lengths less than α'.

By the mean value theorem, there exists $\alpha'' \in (0, \alpha')$ s.t.

$$f(x_k + \alpha' p_k) - f(x_k) = \alpha' \nabla f(x_k + \alpha'' p_k)^T p_k$$

Thus $\nabla f(x_k + \alpha'' p_k)^T p_k = c_1 \underbrace{\nabla f_k^T p_k}_{\leq 0} > c_2 \nabla f_k^T p_k$

Therefore, α'' satisfies the Wolfe conditions.

By the smoothness of f, there is an interval around α'' for which the Wolfe conditions hold.

Since $\nabla f(x_k + \alpha'' p_k)^T p_k < 0$, $\nabla f_k^T p_k < 0$, the strong Wolfe conditions hold.

Backtracking

Choose $\rho \in (0,1)$, $c \in (0,1)$, $\alpha = \bar{\alpha}$

Repeat until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$ → Armijo

$\alpha \leftarrow \rho\alpha$

end

Terminate with $\alpha_k = \alpha$

- $\bar{\alpha} = 1$ for Newton or quasi-Newton
- ρ can vary in $[\rho_{lo}, \rho_{hi}]$ for each step

Global convergence theorem (Zontendijk)

Theorem: Suppose
- p_k is a descent direction
- $x_{k+1} = x_k + \alpha_k p_k$
- α_k satisfies the Wolfe conditions
- f is bounded below
- f is continuously differentiable in an open set N containing the level set $\mathcal{L} = \{x : f(x) \le f(x_0)\}$, x_0 is the starting point
- ∇f is Lipschitz continuously on N, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(\tilde{x})\| \le L\|x - \tilde{x}\| \text{ for all } x, \tilde{x} \in N$$

Then $\sum_{k\ge 0} \cos^2\theta_k \|\nabla f_k\|^2 < \infty$, $\cos\theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\|\|p_k\|}$

This implies $\|\nabla f_k\|^2 \cos^2\theta_k \to 0$

For steepest descent method, $p_k = -\nabla f_k$, $\cos\theta_k = 1$

Then $\|\nabla f_k\| \to 0$

For $p_k = -B_k^{-1}\nabla f_k$ if the condition number of B_k is uniformly bounded by M for all k, it can be shown that $\cos\theta_k \ge \frac{1}{M}$

(condition number: ratio of the largest eigenvalue to the smallest eigenvalue for a positive definite matrix)

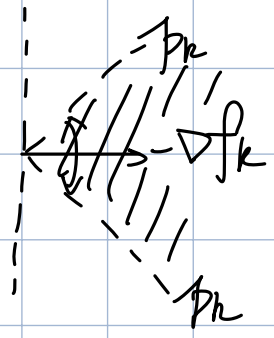
Then $\|\nabla f_k\| \to 0$

So the steepest descent method is globally convergent if α_k satisfies the Wolfe conditions.

Newton method and quasi-Newton methods are globally convergent if α_k satisfies the Wolfe conditions, and B_k has a bounded condition number.

In general, suppose at each iteration $\cos\theta_k \geq \delta > 0$,

then $\lim_{k\to\infty} \|\nabla f_k\| = 0$ (global convergent)



※ Sep. 2nd 2021

How to find p_k?

Consider the first-order appoximation to $f(x)$ about x_k

$$f(x) = \hat{f}(x) = f(x_k) + \nabla f_k^T (x - x_k) \qquad \nabla f_k = \nabla f(x_k)$$

$$\equiv f(x_k) + \nabla f_k^T v_k$$

Maximum decrease in $\hat{f}(x)$ is possible by solving

$$\min_{v_k} \nabla f_k^T v_k \quad \text{s.t.} \quad v_k^T v_k = 1$$

Let θ_k be the angle between ∇f_k and v_k

$$\nabla f_k^T v_k = \|\nabla f_k\| \cdot \|v_k\| \cdot \cos\theta_k$$

$$= \|\nabla f_k\| \cos\theta_k$$

Thus the solution is $v_k = -\nabla f_k / \|\nabla f_k\|$

Steepest Descent Method

· Use the steepest descent direction $p_k = -\nabla f_k$

Algorithm

(1) Initialize x_0 and ε, set $k=0$

(2) While $\|\nabla f_k\| > \varepsilon$

(a) $p_k = -\nabla f_k$

(b) Find $\alpha_k (>0)$ such that — Use exat or backtracking line search

(i) $f(x_k + \alpha_k p_k) < f(x_k)$

(ii) α_k satisfies Wolfe conditions

(c) $x_{k+1} = x_k + \alpha_k p_k$

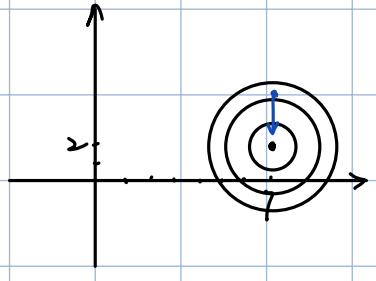
(d) $k = k+1$

end

Output $x^* = x_k$ a stationary point of $f(x)$.

Example. $\min_x f(x)$ $\quad f(x) = (x_1-7)^2 + (x_2-2)^2$

$$\nabla f(x) = \begin{pmatrix} 2(x_1-7) \\ 2(x_2-2) \end{pmatrix} \quad \nabla^2 f(x) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad x^* = \begin{pmatrix} 7 \\ 2 \end{pmatrix}$$

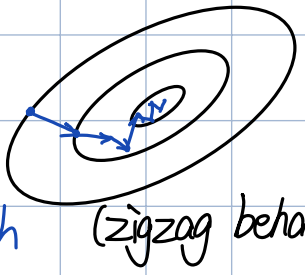


Converge to x^* in one iteration with exact line search

Example. $\min_x f(x)$ $\quad f(x) = 4x_1^2 + x_2^2 - 2x_1x_2$

$$\nabla f(x) = \begin{pmatrix} 8x_1 - 2x_2 \\ 2x_2 - 2x_1 \end{pmatrix} \quad \nabla^2 f(x) = \begin{pmatrix} 8 & -2 \\ -2 & 2 \end{pmatrix} \quad x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

(-1,-2) converge in 26 iterations with exact line search



(zigzag behavior)

(1,0) converge in 4 iterations with exact line search

Convergence of steepest descent method (quadratic case):

Consider the problem

$\min_x f(x)$ where $f(x) = \frac{1}{2}x^TQx - b^Tx$

Q is a symmetric and positive definite matrix

$\nabla f(x) = Qx - b \quad x^* = Q^{-1}b$

How does steepest descent method perform when applied to $f(x)$?

Assum that exact line search is used in each iteration

$$\phi(\alpha_k) = f(x_k - \alpha_k \nabla f_k)$$
$$= \frac{1}{2}(x_k - \alpha_k \nabla f_k)^T Q(x_k - \alpha_k \nabla f_k) - b^T(x_k - \alpha_k \nabla f_k)$$

Set the derivative to 0

$$(Qx_k - Q\alpha_k \nabla f_k - b)^T \nabla f_k = 0$$
$$\Rightarrow (\nabla f_k - Q\alpha_k \nabla f_k)^T \nabla f_k = 0$$
$$\Rightarrow \alpha_k = \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}$$

So $x_{k+1} = x_k - \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k} \nabla f_k$

At what rate does x_k converge?

We introduce the weighted norm $\|x\|_Q^2 = x^TQx$

We want to study $\|x_k - x^*\|_Q^2$

$$\frac{\|x_k - x^*\|_Q^2 - \|x_{k+1} - x^*\|_Q^2}{\|x_k - x^*\|_Q^2}$$

$$= \frac{(x_k - x^*)^TQ(x_k - x^*) - (x_{k+1} - x^*)^TQ(x_{k+1} - x^*)}{(x_k - x^*)^TQ(x_k - x^*)}$$

$(x_{k+1} = x_k - \alpha_k \nabla f_k)$

$$\downarrow = \frac{2\alpha_k \nabla f_k^T \nabla f_k - \alpha_k^2 \nabla f_k^T Q \nabla f_k}{\nabla f_k^T Q^{-1} \nabla f_k}$$

$(\nabla f_k = Qx_k - b = Qx_k - Qx^* = Q(x_k - x^*))$

$$\left(\alpha_k=\frac{\nabla f_k^T\nabla f_k}{\nabla f_k^TQ\nabla f_k}\right)\quad \frac{(\nabla f_k^T\nabla f_k)^2}{(\nabla f_k^TQ\nabla f_k)(\nabla f_k^TQ^{-1}\nabla f_k)}$$

Kantorovich inequality

Let $Q\in\mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Let λ_1 and λ_n be respectively the smallest and the largest eigenvalues

for any $x\neq 0$

$$\frac{(x^Tx)^2}{(x^TQx)(x^TQ^{-1}x)}\geq\frac{4\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2}$$

Using the inequality

$$\frac{\|x_k-x^*\|_Q^2-\|x_{k+1}-x^*\|_Q^2}{\|x_k-x^*\|_Q^2}\geq\frac{4\lambda_1\lambda_n}{(\lambda_1+\lambda_n)^2}$$

Therefore, $\|x_{k+1}-x^*\|_Q^2\leq\left(\frac{\lambda_n-\lambda_1}{\lambda_n+\lambda_1}\right)^2\|x_k-x^*\|_Q^2$

- The steepest descent method converges linearly
- Define the condition number of Q $\underset{(\text{kappa})}{k(Q)}=\frac{\lambda_n}{\lambda_1}$

  Convergence rate of the steepest descent method depends on the condition number of Q
  - $k(Q)=1$ (circular contour) $\Rightarrow$ converge in one iteration
  - $k(Q)\gg 1$ (elliptical contour) $\Rightarrow$ convergence can be slow

  Actual convergence rate depends upon x_0, the starting point.

Sep. 7th. 2021

Nonquadratic functions

Theorem: Suppose that $f\in C^2$, and that the iterates generated by the steepest descent method with exact line searches converges to a point x^* at which the Hessian matrix $\nabla^2 f(x^*)$

(Thm 3.4 in the textbook)

is positive definite. Let r be any scalar in $\left(\frac{\lambda_n-\lambda_1}{\lambda_n+\lambda_1}, 1\right)$ where $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ are eigenvalues of $\nabla^2 f(x^*)$. Then for all k sufficiently large, we have

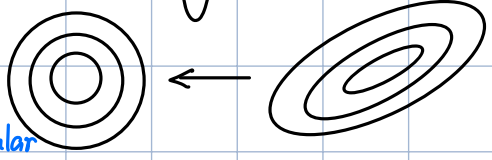
$= 1 - \frac{2}{\kappa(Q)+1}$

$$f(x_{k+1}) - f(x^*) \le r^2 (f(x_k) - f(x^*))$$

Consider the problem to minimize

$$f(x) = \frac{1}{2} x^T Q x - b^T x$$

Fast convergence if $Q = I$



lower triangular

- Let $Q = LL^T$ be the cholesky decomposition of Q
- Let $y = L^T x$

$$f(x) = f(L^{-T} y) = \frac{1}{2} y^T L^{-1} Q L^{-T} y - b^T L^{-T} y$$
$$= \frac{1}{2} y^T y - b^T L^{-T} y \triangleq h(y)$$

Now we apply the steepest descent method to y

$$y_{k+1} = y_k - \nabla h(y_k)$$
$$= y_k - L^{-1} \nabla f(L^{-T} y_k)$$
$$L^{-T} y_{k+1} = L^{-T} y_k - L^{-T} L^{-1} \nabla f(L^{-T} y_k)$$
$$x_{k+1} = x_k - Q^{-1} \nabla f(x_k)$$
$$= x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \longrightarrow \text{Newton's method}$$

Newton's method

How to find p_k?

Consider the second order appoximation to $f(x)$ about x_k

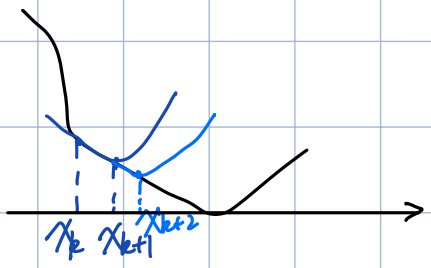
$$f(x) \approx \hat{f}(x) = f(x_k) + \nabla f_k^T (x - x_k) + \frac{1}{2} (x - x_k)^T \nabla^2 f(x_k) (x - x_k)$$

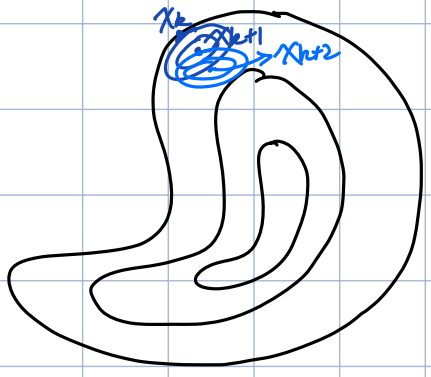
$$x_{k+1} = \arg\min \hat{f}(x)$$

$\nabla \hat{f}(x) = 0 \Rightarrow x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f_k$ (assuming $\nabla^2 f(x_k)$ is invertible)

Classical Newton method

- $p_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$
- $\alpha_k = 1$





Classical Newton algorithm

(1) Initialize x_0 and ε

(2) While $\|\nabla f_k\| > \varepsilon$

(a) $p_k = -\nabla^2 f(x_k)^{-1} \nabla f_k$

(b) $\alpha_k = 1$

(c) $x_{k+1} = x_k + \alpha_k p_k$

(d) $k = k+1$

end

Output: $x^* = x_k$, a stationary point of $f(x)$

- Converges faster than steepest descent method
  ($f(x) = 4x_1^2 + x_2^2 - 2x_1x_2$
  converges in one iteration from any starting point)
- It requires second order information
- Matrix inversion has computing complexity $O(n^3)$ in each iteration
- No guarantee that p_k is a descent direction
  ($\nabla f_k^T p_k = -\nabla f_k^T \nabla^2 f_k \nabla f_k$ not necessarily < 0)
- No guarantee that $f(x_{k+1}) < f(x_k)$ (no line search)

· Sensitive to initial point (for non-quadratic functions)

e.g. $\min_x \log(e^x+e^{-x})$

$$\nabla f(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$$

$$\nabla^2 f(x)=\frac{(e^x+e^{-x})^2-(e^x-e^{-x})^2}{(e^x+e^{-x})^2}$$

$x_0=1 \to x_1=-0.81343 \to x_2=0.4094 \to x_3=-0.0473$

$\to x_4=7.0603\times10^{-5} \to x_5=-2.3455\times10^{-13}$

converges to a local minimizer, which is 0 in this case

$x_0=1.1 \to x_1=-1.12 \to x_2=1.23 \to x_3=-1.695$

$\to x_4=5.715 \to x_5=-23021$

classical Newton algorithm does not converge with this initial value

Def: An iterative optimization algorithm is said to be <u>locally convergent</u> if for each solution x^*, there exists $\delta>0$ such that for any initial point $x_0 \in B(x^*,\delta)$, the algorithm produces a sequence $\{x_k\}$ which converges to x^*.

Theorem: Suppose that $f \in C^2$ and the Hessian $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x^*, where $\nabla f(x^*)=0$ and $\nabla^2 f(x^*)$ is positive definite, then (Thm 3.5 in textbook)

(i) Classical Newton algorithm is locally convergent

(ii) the rate of convergence of $\{x_k\}$ is quadratic

(iii) the sequence of gradient norms $\{\|\nabla f_k\|\}$ converges quadratically to 0.

Pf: $x_{k+1}-x^*=x_k-\nabla^2 f_k^{-1}\nabla f_k-x^*$

$$=\nabla^2 f_k^{-1}[\nabla^2 f_k(x_k-x^*)-(\nabla f_k-\nabla f_*)]$$

($\nabla f_*=\nabla f(x^*)$)

By Taylor's theorem

$$\nabla f_k - \nabla f_* = \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*)\,dt$$

we have

$$\begin{aligned}
&\|\nabla^2 f_k(x_k - x^*) - (\nabla f_k - \nabla f_*)\| \\
&= \|\nabla^2 f_k(x_k - x^*) - \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*)\,dt\| \\
&= \|\int_0^1 [\nabla^2 f_k - \nabla^2 f(x_k + t(x^* - x_k))](x_k - x^*)\,dt\| \\
&\le \int_0^1 \|[\nabla^2 f_k - \nabla^2 f(x_k + t(x^* - x_k))](x_k - x^*)\|\,dt \\
&\le \int_0^1 \|\nabla^2 f_k - \nabla^2 f(x_k + t(x^* - x_k))\| \cdot \|x_k - x^*\|\,dt \\
&= \|x_k - x^*\| \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k))\|\,dt \\
&\le \|x_k - x^*\| \int_0^1 L\|-t(x^* - x_k)\|\,dt \\
&= \|x_k - x^*\| L \|x^* - x_k\| \int_0^1 t\,dt \\
&= \frac{1}{2} L \|x_k - x^*\|^2
\end{aligned}$$

where L is the Lipschitz constant for $\nabla^2 f(x)$ for x and x^*

Since $\nabla^2 f(x^*)$ is nonsingular, there is a radius $r > 0$ such that $\|\nabla^2 f_k^{-1}\| \le 2\|\nabla^2 f(x^*)^{-1}\|$ for all x_k with $\|x_k - x^*\| \le r$.

$$\begin{aligned}
\|x_{k+1} - x^*\| &\le L\|\nabla^2 f(x^*)^{-1}\| \cdot \|x_k - x^*\|^2 \\
&= \tilde{L}\,\|x_k - x^*\|^2
\end{aligned}$$

where $\tilde{L} = L\|\nabla^2 f(x^*)^{-1}\|$

Choosing x_0 such that $\|x_0 - x^*\| \le \min\left(r, \frac{1}{2\tilde{L}}\right)$, we see that

(i) Classical Newton algorithm is locally convergent

(ii) the rate of convergence is quadratic

For (iii), $\|\nabla f(x_{k+1})\| = \|\nabla f(x_{k+1}) - \nabla f_k - \nabla^2 f_k p_k\|$ ($p_k = -\nabla^2 f_k^{-1}\nabla f_k$, $\nabla f_k = -\nabla^2 f_k p_k$)

$$\begin{aligned}
&= \|\int_0^1 \nabla^2 f(x_k + t p_k)(x_{k+1} - x_k)\,dt - \nabla^2 f_k p_k\| \qquad (x_{k+1} - x_k = p_k) \\
&= \|\int_0^1 \nabla^2 f(x_k + t p_k)(x_{k+1} - x_k)\,dt - \nabla^2 f_k p_k\,dt\| \\
&= \|\int_0^1 (\nabla^2 f(x_k + t p_k) - \nabla^2 f(x_k)) p_k\,dt\| \\
&\le \int_0^1 \|\nabla^2 f(x_k + t p_k) - \nabla^2 f(x_k)\| \cdot \|p_k\|\,dt
\end{aligned}$$

$$\leq \|p_k\| \int_0^1 L \|t p_k\| dt$$
$$= \|p_k\|^2 \frac{L}{2}$$
$$= \frac{L}{2} \|\nabla^2 f_k^{-1} \nabla f_k\|^2$$
$$\leq \frac{L}{2} \|\nabla^2 f_k^{-1}\|^2 \cdot \|\nabla f_k\|^2$$
$$\leq 2L \|\nabla^2 f(x^*)^{-1}\|^2 \cdot \|\nabla f_k\|^2$$

Then the gradient norms converges to zero quadratically

If $\|x_0 - x^*\| \leq \min(r, \frac{1}{2L})$, we have locally convergence and quadratic convergence rate, and also quadratic convergence for $\|\nabla f_k\|$

However, it is not practical to check if $\|x_0 - x^*\| \leq \min(r, \frac{1}{2L})$

Initialization of x_0 requires knowledge of x^*.

Modified Newton Method

Issue. $\nabla^2 f_k$ may not be positive definite

Then · $\nabla^2 f_k^{-1}$ may not exist

· p_k may not be [redacted]

Solution: · eigenvalue modification

Let A be a symmetric matrix with spectral decomposition

$A = Q \Lambda Q^T$ where $\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$

Let $\Delta A = Q \begin{pmatrix} \tau_1 & & 0 \\ & \ddots & \\ 0 & & \tau_n \end{pmatrix} Q^T$ with $\tau_i = \begin{cases} 0 & \lambda_i \geq \delta > 0 \\ \delta - \lambda_i & \lambda_i < \delta \end{cases}$

The modified matrix is $A + \Delta A = Q \begin{pmatrix} \lambda_1 + \tau_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n + \tau_n \end{pmatrix} Q^T$

· Add a multiple of the identity matrix

$\nabla^2 f(x_k) + \tau I \leftarrow$ sufficient positive definite

when τ is large, τI is large and the quadratic convergence is lost

· Modified Cholesky decomposition

· Modified symmetric indefinite factorization

Issue: The step length is fix

The convergence is not guaranteed

Solution: Use line search techniques to determine α_k and

$$x_{k+1} = x_k - \alpha_k \nabla^2 f_k^{-1} \nabla f_k$$

Modified Newton Algorithm

(1) Initialize x_0, ε_0 and

(2) While $\|\nabla f_k\| > \varepsilon$

(a) Find the smallest $\tau_k \geq 0$ such that the smallest eigenvalue of $\nabla^2 f_k + \tau_k I$ is greater than δ

(b) Set $p_k = -(\nabla^2 f_k + \tau_k I)^{-1} \nabla f_k$

(c) Find $\alpha_k > 0$ such that α_k satisfies Wolfe conditions

(d) $x_{k+1} = x_k + \alpha_k p_k$

(e) $k = k+1$

end

Output: $x^* = x_k$ a stationary point of $f(x)$

Modified Newton algorithm has global convergence properties and has quadratic convergence rate (Not sensitive to x_0)

Sep. 14th 2021

Quasi-Newton methods (Approximate $\nabla^2 f_k$ by B_k)

Consider the problem $\min_x f(x)$

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function.

Form a quadratic model of f at x_k

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$$

where B_k is a symmetric and positive definite matrix.

Quasi-Newton direction: $p_k = -B_k^{-1} \nabla f_k$

$$x_{k+1} = x_k + \alpha_k p_k = x_k - \alpha_k B_k^{-1} \nabla f_k$$

Given x_k and B_k, how to update B_k to get a symmetric positive definite matrix B_{k+1}?

Suppose we had x_{k+1}, we would construct a new quadratic model

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$$

Require $\nabla m_{k+1}(0) = \nabla f_{k+1}$

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_k \Rightarrow \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$$

Then $B_{k+1} \alpha_k p_k = \nabla f_{k+1} - \nabla f_k$

$$\Leftrightarrow B_{k+1}(x_{k+1} - x_k) = \nabla f_{k+1} - \nabla f_k$$

displacement; change of gradients

Let $y_k = \nabla f_{k+1} - \nabla f_k$, $s_k = x_{k+1} - x_k = \alpha_k p_k$

$$B_{k+1} s_k = y_k \longrightarrow \text{Quasi-Newton condition / secant equation}$$

(n equations, $n(n-1)/2$ unknowns)

Since B_{k+1} should be positive definite

$$s_k^T B_{k+1} s_k = s_k^T y_k > 0$$

Curvature condition

$$\nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k$$

$$\Leftrightarrow (\nabla f_{k+1} - \nabla f_k)^T p_k \geq (c_2 - 1) \nabla f_k^T p_k$$

$$\Leftrightarrow \quad y_k^T p_k \geq \underbrace{(c_2-1)}_{<0}\underbrace{\nabla f_k^T p_k}_{<0} > 0$$

$$\Leftrightarrow \quad y_k^T s_k/\alpha_k > 0 \quad \Leftrightarrow \quad y_k^T s_k > 0$$

When the curvature condition is satisfied, the secant equation always has a solution.

Consider a simple way to update B_k

$B_{k+1} = B_k + \sigma v v^T$ (symmetric rank-1 update / SR1 method)

σ is either 1 or -1, $v \in \mathbb{R}^n$, $v \neq 0$.

$$(B_k + \sigma v v^T) s_k = y_k$$

$$\Rightarrow B_k s_k + (\sigma v^T s_k) v = y_k$$

v tells us

$$v = \delta(y_k - B_k s_k) \text{ for some } \delta$$

$$\Rightarrow y_k - B_k s_k = \sigma\delta^2 [(y_k - B_k s_k)^T s_k](y_k - B_k s_k)$$

$$\Rightarrow \begin{cases} \sigma = \operatorname{sign}[s_k^T (y_k - B_k s_k)] \\ \delta^2 = \dfrac{1}{\sigma[(y_k - B_k s_k)^T s_k]} \end{cases}$$

$$B_{k+1} = B_k + \sigma\delta^2 (y_k - B_k s_k)(y_k - B_k s_k)^T$$

$$= B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

Take $r \in (0,1)$ to be a small number, say $r = 10^{-8}$

If $|(y_k - B_k s_k)^T s_k| < r \|s_k\| \cdot \|y_k - B_k s_k\|$

then set $B_{k+1} = B_k$

In case the denominator is too small.

Update the inverse "Hessian" matrix (Apply Sherman-Morrison formula)

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}$$

Quasi-Newton algorithm (SR1 method)

(1) Initialize x_0, ε and symmetric positive definite matrix B_0 (say I_n), set $k = 0$

(2) While $\|\nabla f_k\| > \varepsilon$

(a) $p_k = -H_k \nabla f_k$

(b) Find α_k satisfying Wolfe conditions

(c) $x_{k+1} = x_k + \alpha_k p_k$

(d) Find H_{k+1} using SR1 method

(e) $k = k+1$

end

Output: $x^* = x_k$ a stationary point of $f(x)$.

~~7~~ Sep. 16th 2021

R: crossprod(A) → A^TA

tcrossprod(A) → AA^T

Ex. $\min_x f(x)$ $f(x) = 4x_1^2 + x_2^2 - 2x_1x_2$

inexact line search

· $x_0 = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$

k	x_1	x_2	H_k	$\|\nabla f_k\|$
0	-2	-2	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	12
1	-0.92	-2	$\begin{pmatrix} 0.183 & 0.233 \\ 0.233 & 0.933 \end{pmatrix}$	3.99
2	0.2	0.8	$\begin{pmatrix} 0.167 & 0.167 \\ 0.167 & 0.667 \end{pmatrix}$	1.2
3	0	0	$\begin{pmatrix} 0.167 & 0.167 \\ 0.167 & 0.667 \end{pmatrix}$	0

· $x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

k	x_1	x_2	H_k	$\|\nabla f_k\|$
0	1	0	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	8.246
1	0.28	0.18	$\begin{pmatrix} 0.189 & 0.243 \\ 0.243 & 0.927 \end{pmatrix}$	1.89
2	-0.027	-0.092	$\begin{pmatrix} 0.167 & 0.167 \\ 0.167 & 0.667 \end{pmatrix}$	0.134
3	0	0	$\begin{pmatrix} 0.167 & 0.167 \\ 0.167 & 0.667 \end{pmatrix}$	0

Property of SR1 method

Thm: $\min_x f(x)$ $f(x) = \frac{1}{2}x^TAx + b^Tx$ $(A>0)$

For simplicity, $x_{k+1} = x_k + p_k$ $p_k = -H_k \nabla f_k$

Assume $(s_k - H_k y_k)^T y_k \neq 0$ for all k.

Then for any starting value H_0, where H_0 is symmetric, we have $\{x_k\}$ converges to the minimizer in at most $n+1$ steps.

If the p_i's are linearly independent, then $H_n = A^{-1}$.

Pf: First we show that $H_n y_j = s_j$, $j = 0, \cdots, k-1$ (chereditary property)

Suppose we have hereditary property, then

$$(s_0, s_1, \cdots, s_{n-1}) = (H_n y_0, H_n y_1, \cdots, H_n y_{n-1})$$
$$= H_n A (s_0, s_1, \cdots, s_{n-1})$$

$y_i = \nabla f_{i+1} - \nabla f_i$
$= (Ax_{i+1} + b) - (Ax_i + b)$
$= A(x_{i+1} - x_i)$
$= As_i$

$p_i = x_{i+1} - x_i = s_i$

Because p_i's are linearly independent, s_i's are linearly independent.

We have $(s_0, s_1, \cdots, s_{n-1})$ being a nonsingular matrix

Thus $H_n A = I_n \Rightarrow H_n = A^{-1}$

Now we prove the hereditary property

It is shown by induction

By the secant equation, we have $H_1 y_0 = s_0$

Assume that this holds for some $k \geq 1$, then (i.e., $H_k y_j = s_j$)

$$(s_k - H_k y_k)^T y_j = s_k^T y_j - y_k^T H_k y_j$$
$$= s_k^T y_j - y_k^T s_j$$
$$= s_k^T A s_j - s_k^T A s_j = 0$$

$$H_{k+1} y_j = \left[H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \right] y_j$$

(the bracketed term $= 0$)

For $j = k$, $H_{k+1} y_k = s_k$ because of □.

7: Sep. 21st 2021

(proof continued)

If the steps becomes linearly dependent. Suppose that s_k is a linear combination of the previous steps, i.e.,

$$s_k = \xi_0 s_0 + \xi_1 s_1 + \cdots + \xi_{k-1} s_{k-1}$$

for some scalar ξ_i, $i = 1, 2, \cdots, k-1$

$$H_k y_k = H_k A s_k$$

$$= H_k A(\xi_0 s_0 + \xi_1 s_1 + \cdots + \xi_{k-1} s_{k-1})$$
$$= \xi_0 H_k y_0 + \xi_1 H_k y_1 + \cdots + \xi_{k-1} H_k y_{k-1}$$
$$= H_k(\xi_0 y_0 + \xi_1 y_1 + \cdots + \xi_{k-1} y_{k-1})$$
$$= \xi_0 s_0 + \xi_1 s_1 + \cdots + \xi_{k-1} s_{k-1}$$
$$= s_k$$

Since $y_k = \nabla f_{k+1} - \nabla f_k$, and $s_k = p_k = -H_k \nabla f_k$

$$H_k(\nabla f_{k-1} - \nabla f_k) = -H_k \nabla f_k$$

This implies $H_k \nabla f_{k+1} = 0 \Rightarrow \nabla f_{k+1} = 0$

So x_{k+1} is the solution

Thm 6.2 of the textbook

Theorem: Suppose that $f \in C^2$. Its Hessian is bounded and Lipschitz continuous in a neighborhood of x^*. Assume $x_k \to x^*$, and $|s_k^T(y_k - B_k s_k)| \geq r\|s_k^T\| \|y_k - B_k s_k\|$ for some $r \in (0,1)$, and s_k are uniformly linearly independent.

Then $\lim_{k\to\infty} \|B_k - \nabla^2 f(x^*)\| = 0$.

Thm 3.6 of the textbook

Theorem: For a general line search method, $x_{k+1} = x_k + \alpha_k p_k$

√ · $f \in C^2$

· α_k satisfies the Wolfe conditions

· p_k is a descent direction

√√ · $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)$ is positive definite

· If $\lim_{k\to\infty} \frac{\|\nabla f_k + \nabla^2 f_k p_k\|}{\|p_k\|} = 0$

then (i) the step length $\alpha_k = 1$ is admissible for all k greater than a certain index k_0

(ii) if $\alpha_k = 1$ for all $k > k_0$, $\{x_k\}$ converges to x^* superlinearly

If p_k is a quasi-Newton search direction, $x_{k+1} = x_k + p_k$ and

$\checkmark + \checkmark$, $x_k \to x^*$, $\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|} = 0$ is a necessary and sufficient condition for superlinear convergence.

Pf. Let $p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$ be the Newton step

$$\begin{aligned} p_k - p_k^N &= \nabla^2 f_k^{-1}(\nabla^2 f_k p_k + \nabla f_k) \\ &= \nabla^2 f_k^{-1}(\nabla^2 f_k - B_k)p_k \qquad (p_k = -B_k^{-1}\nabla f_k) \\ &= O(\|(\nabla^2 f_k - B_k)p_k\|) \\ &= o(\|p_k\|) \end{aligned}$$

Now if we have $p_k - p_k^N = o(\|p_k\|)$, multiply both sides by $\nabla^2 f_k$

$$\nabla^2 f_k(p_k + \nabla^2 f_k^{-1}\nabla f_k) = o(\|\nabla^2 f_k p_k\|) = o(\|p_k\|)$$

$$\nabla^2 f_k p_k + \nabla f_k = o(\|p_k\|)$$

$$(\nabla^2 f_k - B_k)p_k = o(\|p_k\|)$$

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f_k)p_k\|}{\|p_k\|} = 0 \text{ , i.e., } \lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|} = 0$$

Thus $\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^*))p_k\|}{\|p_k\|} = 0$ is equivalent to $p_k - p_k^N = o(\|p_k\|)$

$$\begin{aligned} \|x_{k+1} - x^*\| &= \|x_k + p_k - x^*\| \\ &\le \|x_k + p_k^N - x^*\| + \|p_k - p_k^N\| \\ &= O(\|x_k - x^*\|^2) + o(\|p_k\|) \end{aligned}$$

contradiction

- if $\|p_k\| = o(\|x_k - x^*\|)$, $\|x_k + p_k - x^*\| = O(\|x_k - x^*\|)$
- if $\|x_k - x^*\| = o(\|p_k\|)$, $\|x_k + p_k - x^*\| = O(\|p_k\|)$

Thus $\|p_k\| = O(\|x_k - x^*\|)$

So we obtain $\|x_k + p_k - x^*\| \le o(\|x_k - x^*\|)$,

i.e., superlinear convergence.

Drawbacks of SR1 method

- B_{k+1} is positive definite if $(y_k - B_k s_k)^T s_k > 0$, which cannot be guaranteed

· Numerical difficulties if $(y_k - B_k s_k)^T s_k \approx 0$

Rank Two methods

(The following methods have received wide acceptance)

· Davidon-Fletcher-Powell (DFP) method

· Broydon-Fletcher-Goldfarb-Shanno (BFGS) method

Rank one:

$B_{k+1} s_k = y_k$ equivalently $H_{k+1} y_k = s_k$

where $H_{k+1} = H_k + \alpha u u^T$

Now suppose that H_k is symmetric and positive definite. Let

$H_{k+1} = H_k + \alpha u u^T + \beta v v^T$, $\alpha \neq 0, \beta \neq 0, u \neq 0, v \neq 0$

$H_{k+1} y_k = s_k$

$\Rightarrow H_k y_k + \alpha u u^T y_k + \beta v v^T y_k = s_k$

$\Rightarrow \alpha u u^T y_k + \beta v v^T y_k = s_k - H_k y_k$

Let $\alpha = \frac{1}{s_k^T y_k}$, $\beta = -\frac{1}{y_k^T H_k y_k}$ $(u = s_k, v = -H_k y_k)$

Therefore $H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k^T}{y_k^T H_k y_k}$ (DFP method)

$\sim n^2$ $\sim n^2$ $\sim n^2$ $\sim 2n^2 + n^2 = O(6n^2)$

$B_{k+1} = H_{k+1}^{-1}$ is the solution to the following problem

$\|A\|_F = \sqrt{tr(A^T A)}$

$$\min_B \| W^{1/2}(B - B_k) W^{1/2} \|_F \quad \text{subject to } B = B^T, \ B s_k = y_k$$

W is given in (6.11) of the textbook, W is symmetric and $W y_k = s_k$

$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T$, where $\rho_k = \frac{1}{y_k^T s_k}$

△ If H_k is positive definite, is H_{k+1} positive definite?

Let $x \in \mathbb{R}^n$, $x \neq 0$

$$x^T H_{k+1} x = x^T H_k x + \frac{x^T s_k s_k^T x}{s_k^T y_k} - \frac{x^T H_k y_k y_k^T H_k x}{y_k^T H_k y_k}$$

Let $a = H_k^{1/2} x$ and $b = H_k^{1/2} y_k$

$$x^T H_{k+1} x = a^T a - \frac{(a^T b)(b^T a)}{b^T b} + \frac{(s_k^T x)^2}{s_k^T y_k}$$

$$= \frac{(a^T a)(b^T b) - (a^T b)^2}{b^T b} + \frac{(s_k^T x)^2}{s_k^T y_k} \geq 0$$

($(a^T a)(b^T b) - (a^T b)^2 \geq 0$ by C-S; $(s_k^T x)^2 \geq 0$; $s_k^T y_k > 0$ because of Wolfe conditions)

Wolfe conditions:

$$\nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k, \quad 0 < c_2 < 1$$

$$(\nabla f_{k+1}^T - \nabla f_k^T) p_k \geq (c_2 - 1) \nabla f_k^T p_k$$

$$y_k^T s_k / \alpha_k \geq (c_2 - 1) \nabla f_k^T p_k > 0$$

($y_k^T s_k = \alpha_k y_k^T p_k$; $c_2 - 1 < 0$; $\nabla f_k^T p_k < 0$)

Thus $x^T H_{k+1} x \geq 0$

and the equality holds if and only if

$$(a^T a)(b^T b) = (a^T b)^2 \quad \text{and} \quad x^T s_k = 0$$

(Notice $a \neq 0$, $b \neq 0$, which can be proved by $H_k > 0$ and $s_k^T y_k > 0$)

Then $a = \lambda b \Rightarrow x = \lambda y_k$, $\lambda \neq 0$

$$x^T s_k = 0 \Rightarrow \lambda y_k^T s_k = 0 \Rightarrow \lambda = 0 \quad \text{contradiction}$$

($s_k^T y_k > 0 \Rightarrow y_k \neq 0$)

Thus H_{k+1} is positive definite.

BFGS method solves

$$\min_H \| W^{1/2} (H - H_k) W^{1/2} \|_F \quad \text{subject to } H = H^T, \; H y_k = s_k$$

W is symmetric and $W s_k = y_k$.

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \quad \text{where } \rho_k = \frac{1}{s_k^T y_k}$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

$O(9n^2)$

BFGS problem is a dual problem to the DFP method

Compared to DFP method, BFGS method is more efficient in self-correction. (If H_k incorrectly estimates the curvature of the objective function, the BFGS formula will tend to correct itself within a few steps)

7 Sep. 23rd 2021

Quasi-Newton Algorithm (Rank 2 method)

(1) Initialized x_0, ε and symmetric positive definite matrix H_0, set $k=0$.
($H_0=I$, run one iteration and then $H_0=\frac{y_k^T s_k}{s_k^T s_k} I$)

(2) While $\|\nabla f_k\|>\varepsilon$

(a) $p_k=-H_k\nabla f_k$

$\left(\frac{\|\nabla f_k+\nabla^2 f_k p_k\|}{\|p_k\|}=\frac{\|-B_k p_k+\nabla^2 f_k p_k\|}{\|p_k\|}\le \|B_k-\nabla^2 f_k\|\to 0\right)$ ↑ thm 6.2 & 3.6 ∵

(b) Find α_k satisfying Wolfe conditions (try $\alpha_k=1$ first)

(c) $x_{k+1}=x_k+\alpha_k p_k$

(d) Find H_{k+1} using DFP/BFGS method ← $O(n^2)$

if $y_k^T s_k$ is close to 0, apply damped BFGS update (chapter 18 of the textbook page 537)

(e) $k=k+1$

end

Example. $\min_x f(x)$ $f(x)=100(x_2-x_1)^2+(1-x_1)^2$

methods	steepest descent	BFGS	Newton
iterations	5273	32	19

The Broyden class

$$B_{k+1}=(1-\phi_k)B_{k+1}^{BFGS}+\phi_k B_{k+1}^{DFP}$$ (ϕ_k: Broyden parameter)

$$\phi_k^c=\frac{1}{1-\mu_k} \qquad \mu_k=\frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}\ge 1$$

B_{k+1} is positive definite (for all $\phi_k>\phi_k^c$) and satisfies Quasi-Newton condition

$$\phi_k=\frac{s_k^T y_k}{s_k^T y_k-s_k^T B_k s_k}\to SR-1$$

Restricted Broyden class $\phi_k\in[0,1]$

$\phi_k=0 \to$ BFGS

$\phi_k=1 \to$ DFP

Theorem $f(x)=\frac{1}{2}x^T A x+b^T x$ A is symmetric positive definite

x_0 any starting point

B_0 symmetric positive definite

$p_k = -B_k^{-1}\nabla f_k$ $\qquad x_{k+1} = x_k + p_k$

using Broyden formula

Let $\lambda_1^k \le \lambda_2^k \le \cdots \le \lambda_n^k$ be the eigenvalues of $A^{1/2} B_k^{-1} A^{1/2}$

Then for all k, we have

$$\min\{\lambda_i^k, 1\} \le \lambda_i^{k+1} \le \max\{\lambda_i^k, 1\}, \quad i = 1, 2, \cdots, n$$

Moreover, this does not hold if ϕ_k is chosen outside $[0, 1]$.

Theorem. $f(x) = \frac{1}{2} x^T A x + b^T x$ $\quad A$ is symmetric positive definite

x_0 any starting point

B_0 symmetric positive definite

α_k the exact step length (exact line search)

$\phi_k > \phi_k^c$ for all k

Then (1) The iterates converge to the solution in at most $n+1$ steps regardless of ϕ_k

(2) $B_n = A$

(3) The secant equation is satisfied for all previous search directions $B_k s_j = y_j$

Global convergence of BFGS method

Assumptions: (i) $f \in C^1$

(ii) The level set $\mathcal{L} = \{x \in \mathbb{R}^n \mid f(x) \le f(x_0)\}$ is convex, and there exists positive constants m and M such that

$m\|z\|^2 \le z^T \nabla^2 f(x) z \le M\|z\|^2$ for all $z \in \mathbb{R}^n$ and $x \in \mathcal{L}$

Theorem: Assume Assumption (i) and (ii) hold, B_0 is a symmetric positive definite matrix. Then $\{x_k\}$ converges to the minimizer x^* of f, where x_k is generated from the BFGS method.

This result has been extended to the entire restricted Broyden class, except for the DFP method.

Assumption (iii) The Hessian matrix is Lipschitz continuous at x^*,

i.e., $\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \le L\|x - x^*\|$ for all x near x^*, $L > 0$

Theorem:
- $f \in C^2$
- $\{x_k\}$ generated by the BFGS method algorithm converge to a minimizer x^* at which Assumption (ii) holds
- $\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty$

Then x_k converges to x^* at a superlinear rate

7: Sep. 28th 2021

Stochastic gradient descent (SGD)

Bertsekas (2010) "Incremental gradient, subgradient, and proximal methods for convex optimization: a survey"

Nemirouski, Juditsky, Lan and Shapiro (2009) "Robust stochastic optimization approach to stochastic programming"

Consider minimizing an average of functions

$$\min_{x \in \mathbb{R}^n} f(x) \qquad f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$$

As $\nabla \sum_{i=1}^{m} f_i(x) = \sum_{i=1}^{m} \nabla f_i(x)$, the steepest descent method would have (gradient descent)

the following formula

$$x_{k+1} = x_k - \alpha_k \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x_k)$$

In comparison, SGD (or incremental gradient descent) has the updating formula

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k)$$

$\nabla f_{i_k}(x_k)$ may not be a descent direction, so we cannot perform backtracking line search.

where $i_k \in \{1, \cdots, m\}$ is some chosen index at iteration $k+1$.

Two rules for choosing index i_k:

- Randomized rule: Choose $i_k \in \{1, \cdots m\}$ uniformly at random
- Cyclic rule: Choose $i_k = 1, 2, \cdots, m, 1, 2, \cdots, m, \cdots$

Randomized rule is more common in practice. For randomized rule, note that $E[\nabla f_{i_k}(x)] = \nabla f(x)$

so we can view SGD as using an unbiased estimate of the gradient at each step.

Main appeal of SGD:

- Iteration cost is independent of m
- Can also be a big savings in terms of memory usage

Example: Logistic regression

Given $(x_i, y_i) \in \mathbb{R}^p \times \{0, 1\}$, $i = 1, \cdots, m$, recall the logistic regression solves $\min_{\beta \in \mathbb{R}^p} f(\beta)$ $f(\beta) = -\frac{1}{m} \sum_{i=1}^{m} \underbrace{[y_i x_i^T \beta - \log(1 + e^{x_i^T \beta})]}_{f_i(\beta)}$

Gradient $\nabla f(\beta) = -\frac{1}{m} \sum_{i=1}^{m} (y_i - \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}}) x_i$

Full gradient (also called batch) versus stochastic gradient

- One batch update costs $O(mp)$
- One stochastic update costs $O(p)$

Rule of thumb for stochastic method

- generally thrive far from optimum
- generally struggle close to optimum

Step length: Standard in SGD is to use diminishing step length

e.g. $\alpha_k = \frac{1}{k}$ for $k = 1, 2, 3, \cdots$

Why not fixed step length? Here is some intuition. Suppose we take cyclic rule for simplicity. Set $\alpha_k = \alpha$ for m updates in a row, we get $x_{k+1} = x_k - \alpha \nabla f_1(x_k)$

$$x_{k+2} = x_{k+1} - \alpha \nabla f_2(x_{k+1})$$
$$= x_k - \alpha \nabla f_1(x_k) - \alpha \nabla f_2(x_{k+1})$$
$$\vdots$$
$$x_{k+m} = x_k - \alpha \nabla f_1(x_k) - \alpha \nabla f_2(x_{k+1}) - \cdots - \alpha \nabla f_m(x_{k+m-1})$$
$$= x_k - \alpha \sum_{i=1}^{m} \nabla f_i(x_{k+i-1})$$

Meanwhile, full gradient with step length $m\alpha$ would give

$$x_{k+1} = x_k - \alpha \sum_{i=1}^{m} \nabla f_i(x_k)$$

The difference here is $\alpha \left[\sum_{i=1}^{m} \nabla f_i(x_{k+i-1}) - \sum_{i=1}^{m} \nabla f_i(x_k) \right]$

If we hold α constant, this difference will not generally be going to 0.

Convergence rate

Theorem: Assume that $E(\|\nabla f_i\|^2) \le G^2$ and f is strongly convex, i.e., there exists a positive number $l > 0$ such that $f - \frac{l}{2}\|x\|^2$ is convex.

Choose the step length $\alpha_k = \frac{1}{lk}$. We have

$$E(\|x_k - x^*\|^2) \le \frac{\max\{\|x_1 - x^*\|^2, G^2/l^2\}}{k}$$

where x^* is the optimal solution.

Another form of convergence rate:

If f is convex, the steepest descent method which diminishing step sizes satisfies $f(x_k) - f(x^*) = O(\frac{1}{\sqrt{k}})$

When f has Lipschitz gradient, we get for suitable fixed step sizes $f(x_k) - f(x^*) = O(\frac{1}{k})$

What about SGD?

If f is convex, SGD with diminishing step sizes satisfies

$$E(f(x_k)) - f(x^*) = O(\frac{1}{\sqrt{k}})$$

Unfortunately, this does not improve when we further assume f has Lipschitz gradient.

Even worse is the following discrepancy.

When f is strongly convex, and has a Lipschitz gradient, gradient descent satisfies $f(x_k) - f(x^*) = O(c^k)$ where $c < 1$

But under same conditions, SGD gives us

$$E(f(x_k)) - f(x^*) = O(\frac{1}{k})$$

So SGD does not enjoy the linear convergence rate of gradient descent under strong convexity.

What can we do to improve SGD?

Also common is mini-batch SGD, where we choose a random subset $I_k \subseteq \{1, \dots, m\}$ of size $|I_k| = b << m$, and repeat

$$x_{k+1} = x_k - \alpha_k \frac{1}{b} \sum_{i \in I_k} \nabla f_i(x_k)$$

Again we are approximating full gradient by an unbiased estimate

$$E[\frac{1}{b} \sum_{i \in I_k} \nabla f_i(x)] = \nabla f(x)$$

Using mini-batches reduces the variance of the gradient estimate by a factor $\frac{1}{b}$, but is also b times more expensive computationally.

Back to the logistic regression example, let's consider a regularized version $\min_{\beta \in \mathbb{R}^p} f(\beta)$

$$f(\beta) = -\frac{1}{m} \sum_{i=1}^m (y_i x_i^T \beta - \log(1 + e^{x_i^T \beta})) + \frac{\lambda}{2} \|\beta\|_2^2$$

$$\stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^m f_i(\beta)$$

Full gradient computation is $\nabla f(\beta) = \frac{-1}{m} \sum_{i=1}^m (y_i - \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}}) x_i + \lambda \beta$

Comparison between methods

- One batch update costs $O(np)$
- One mini-batch update costs $O(bp)$
- One stochastic update costs $O(p)$

7: Sep 20th 2021

SGD can be super efficient in terms of iteration cost, memory

But SGD is slow to converge, cannot adapt to strong convexity

f	GD	SGD
convex	$\frac{1}{\sqrt{k}}$	$\frac{1}{\sqrt{k}}$
+ Lipschitz gradient	$\frac{1}{k}$	$\frac{1}{\sqrt{k}}$
+ strong convexity	c^k	$\frac{1}{k}$

- Mini-batch may be useful in practice

For a while, slow convergence for strongly convex functions was believed inevitable as Nemirovski and others established matching lower bounds, but this was for a more general stochastic problem. where $f(x) = \int F(x, \xi) dP(\xi)$.

New way of "variance reduction" work shows we can modify SGD to converge much faster for finite sums.

SGD has really taken off in large-scale machine learning.

- In many machine learning problems we do not care about optimizing in high accuracy, it does not pay off in terms of statistical performance.
- Thus (in contrast to what classic theory says) fixed step sizes

are commonly used in machine learning application.

- One trick is to experiment with step lengths using small fraction of training before running SGD on full data set... many other heuristics are common

** Bothou (2012) "Stochastic gradient descent tricks"

- Many variants provide better stability, convergence.

  SVRG, SAG, SAGA, AdaGrad, Adam, AdaMax, ...

△ Stochastic average gradient (SAG)

** Schmidt, Le Roux and Bach (2013) "Minimizing finite sums with the stochastic average gradient"

SAG is a breakthrough method in stochastic optimization

- Maintain table, containing g_i of f_i, $i=1,\cdots,m$
- Initialize $x^{(0)}$ and $g_i^{(0)} = \nabla f_i(x^{(0)})$, $i=1,\cdots,m$
- At step $k=1,2,3,\cdots$, pick random $i_k \in \{1,\cdots,m\}$, then let

  $g_{i_k}^{(k)} = \nabla f_{i_k}(x^{(k-1)})$ (most recent gradient of f_{i_k})

  Set all other $g_i^{(k)} = g_i^{(k-1)}$ $i \neq i_k$, i.e., these stay the same
- Update $x^{(k)} = x^{(k-1)} - \alpha_k \frac{1}{m} \sum_{i=1}^{m} g_i^{(k)}$

The key of SAG is to allow each f_i, $i=1,\cdots,n$ to communicate a part of the gradient estimate at each step. The basic idea can be traced back to incremented aggregated gradient (Blatt, Hero, Gauchman, 2006)

SAG gradient estimates are no longer unbiased, but they have greatly reduced variance.

Computationally, SAG is as efficient as SGD

$$x^{(k)} = x^{(k-1)} - \alpha_k \left(\frac{g_{ik}^{(k)}}{m} - \frac{g_{ik}^{(k-1)}}{m} + \underbrace{\frac{1}{m}\sum_{i=1}^{m} g_i^{(k-1)}}_{\text{old table average}} \right) \rightarrow \text{new table average}$$

7-Oct. 5th 2021

Convergence analysis

Assume that $f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$, where each $f_i \in D$, and ∇f is Lipschitz with constant L.

Denote $\bar{x}^{(k)} = \frac{1}{k}\sum_{\ell=0}^{k-1} x^{(\ell)}$, the average iterate after $k-1$ steps

Theorem (Schmidt et al. 2013) SAG with a fixed step size $\alpha = \frac{1}{16L}$, and the initialization $g_i^{(0)} = \nabla f_i(x^{(0)}) - \nabla f(x^{(0)})$, $i = 1, \dots, m$ satisfies

$$E[f(\bar{x}^{(k)})] - f(x^*) \leq \frac{48m}{k}(f(x^{(0)}) - f(x^*)) + \frac{128L}{k}\|x^{(0)} - x^*\|_2^2$$

where the expectation is taken over random choices of indices.

- The result is stated in terms of the average iterate $\bar{x}^{(k)}$, but also can be shown to hold for best iterate $x_{best}^{(k)}$ seen so far.
- The convergence rate $O(\frac{1}{k})$ is comparable to gradient descent, better than $O(\frac{1}{\sqrt{k}})$ for SGD
- First term in SAG bound suffers from factor of m. The authors suggest smarter initialization to make $f(x^{(0)}) - f(x^*)$ small (e.g. take one full cycle of SGD to get $x^{(0)}$ and set $g_i^{(0)} = \nabla f_i(x^{(0)})$, and use $x^{(0)}$ to start SAG. This warm start helps a lot.

Theorem (Schmidt et al. 2013) Assume further that each f_i is strongly convex with parameter ℓ. Then SAG with a step size $\alpha = \frac{1}{16L}$ and the same initialization as before, satisfies

$$E[f(x^{(k)})] - f(x^*) \leq \left(1 - \min\left\{\frac{\ell}{16L}, \frac{1}{8m}\right\}\right)^k \left(\frac{3}{2}(f(x^{(0)}) - f(x^*)) + \frac{4L}{m}\|x^{(0)} - x^*\|_2^2\right)$$

- This is linear convergence rate $O(c^k)$ for SAG. Compare this to

$O(c^k)$ for GD and only $O(\frac{1}{k})$ for SGD

- Tuning the fixed step sizes for SAG was tricky
- Authors of SAG conveyed that this algorithm will work the best, relative to SGD, for ill-conditioned problems.

SAGA

** Defazio, Bach and Locoste-Julien (2014) "SAGA: A fast incremented gradient method with support for non-strongly convex composite objectives"

SAGA is a follow-up on the SAGA work

- Maintain table, containing gradient g_i of f_i, $i=1,\cdots,m$
- Initialize $x^{(0)}$, and $g_i^{(0)} = \nabla f_i(x^{(0)})$, $i=1,\cdots,m$
- At steps $k=1,2,3,\cdots$, pick random $i_k \in \{1,\cdots,m\}$, then let $g_{i_k}^{(k)} = \nabla f_{i_k}(x^{(k-1)})$ set all other $g_i^{(k)} = g_i^{(k-1)}$, $i \neq i_k$, i.e., these stay the same
- Update $x^{(k)} = x^{(k-1)} - \alpha_k (g_{i_k}^{(k)} - g_{i_k}^{(k-1)} + \frac{1}{m}\sum_{i=1}^m g_i^{(k-1)})$

SAGA gradient estimate $g_{i_k}^{(k)} - g_{i_k}^{(k-1)} + \frac{1}{m}\sum_{i=1}^m g_i^{(k-1)}$

v.s.

SAG gradient estimate $\frac{1}{m} g_{i_k}^{(k)} - \frac{1}{m} g_{i_k}^{(k-1)} + \frac{1}{m}\sum_{i=1}^m g_i^{(k-1)}$

- Recall SAG estimate is biased, remarkably, SAGA estimate is unbiased!

Simple explanation: Consider family of estimators $\theta_\alpha = \alpha(X-Y)+E(Y)$ for $E(X)$, where $\alpha \in [0,1]$, and X, Y are presumed correlated.

We have $E(\theta_\alpha) = \alpha E(X) + (1-\alpha)E(Y)$

$$Var(\theta_\alpha) = \alpha^2(Var(X) + Var(Y) - 2Cov(X,Y))$$

SAGA uses $\alpha = 1$ (unbiased), SAG uses $\alpha = \frac{1}{n}$ (biased)

- SAGA matches convergence rates of SAG
- Initialize at $g_i^{(0)} = \nabla f_i(x^{(0)})$, where $x^{(0)}$ is obtained by taking a full cycle of SGD
- SAGA curves look more like SGD curves (jagged and highly variable), SAG

updates have much lower variance.

△ Coordinate Descent Method (Section 9.3)

Consider the problem $\min_x f(x)$ where $f: \mathbb{R}^n \to \mathbb{R}$, $f \in C^1$, $x = (x_1, \cdots, x_n)^T$

Idea ① For every coordinate viable x_i, $i = 1, \cdots, n$, minimize $f(x)$ w.r.t. x_i, keeping the other coordinate variables x_j, $j \neq i$ constant

② Repeat the procedure in ① until stopping criterion is satisfied

- Given convex, differentiable $f: \mathbb{R}^n \to \mathbb{R}$ if we are at a point x s.t. $f(x)$ is minimized along each coordinate axis, then have we found a global minimizer? $(f(x_i + \delta e_i) \geq f(x_i),\ \forall \delta, i,\ e_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow i\text{th place})$

  Yes, because $0 = \nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n}\right) = 0$.

- Same question, how about f convex, but not differentiable?

  No. counterexample

- Same question again, how about $f(x) = g(x) + \sum_{i=1}^{n} h_i(x_i)$, where g convex, differentiable and each h_i convex? (Here the nonsmooth part is called seperable)

  Yes

  Algorithm

  (1) Initialize x_0, ε and $k = 0$

  (2) While $\|\nabla f^{(k)}\| > \varepsilon$

  for $i = 1, \cdots, n$

  $$x_i^{(k)} = \operatorname{argmin}_{x_i} f(x_1^{(k)}, \cdots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, \cdots, x_n^{(k-1)})$$

  end

end

Output $x^* = x^{(k)}$, a stationary point of $f(x)$

Important note: We always use most recent information possible.

Global convergent method if a search along any coordinate direction yields a unique minimum point.

Tseng (2001) proves that for $f(x) = g(x) + \sum_{i=1}^{n} h_i(x_i)$ with g convex and differentiable and each h_i convex (provided f is continuous on a compact set $\{x : f(x) \le f(x^{(0)})\}$ and f attains its minimum), any limit point of $x^{(k)}$ is a minimizer of f.

Notes: · Order of cycle through coordinate is arbitrary, can use any permutation of $\{1, 2, \cdots, n\}$

· Can replace individual coordinates with blocks of coordinates

· "One-at-a-time" update is critical, and "all-at-once" scheme does not necessarily converge.

Z: Oct. 7th 2021

Example. Linear regression

Given $Y \in \mathbb{R}^m$ and $X \in \mathbb{R}^{m \times p}$ with columns $X_1, \cdots, X_p$, consider the LS problem

$$\min_{\beta} \frac{1}{2} \|Y - X\beta\|_2^2$$

Minimizing over β_i, with all β_j, $j \neq i$ fixed

$$0 = \nabla_i f(\beta) = X_i^T (X\beta - Y) = X_i^T (X_i \beta_i + X_{-i} \beta_{-i} - Y)$$

$\beta_{-i} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_{i-1} \\ \beta_{i+1} \\ \vdots \\ \beta_p \end{pmatrix}$

We take $\beta_i = \dfrac{X_i^T (Y - X_{-i} \beta_{-i})}{X_i^T X_i}$

Coordinate descent repeat this update for $i = 1, \cdots, p, 1, 2, \cdots, p, \cdots$

· Note that this is exactly Gauss-Seided for the system $X^T X \beta = X^T Y$

History in statistics / machine learning

- Idea appeared in Fu (1998), and then again in Daubechies et al. (2004), but was inexplicably ignore.
- Later, three papers in 2007, especially Friedman et al. (2007), really sparked interest in statistics and ML communities

Why is it used?

- Very simple and easy to implement
- Careful implementation can achieve state-of-the-art
- Scalable, e.g. do not need to keep full data in memory

Examples: Lasso regression (R package glmnet), lasso GLMs, SVMs, group lasso, graphical lasso (applied to the dual), additive modeling, matrix completion, regression with nonconvex penalties.

Roughly speaking, the convergence rates are similar to those for gradient descent.

Jacobi and Gauss-Seidel methods for solving linear systems.

Suppose $A \succ 0$. Jacobi and Gauss-Seidel are two basic iterate approaches for solving $Ax = b$.

△ Jacobi iterations:

Initialize $x^{(0)} \in \mathbb{R}^n$, repeat for $k = 1, 2, 3, \cdots$,

$$x_i^{(k)} = \left(b_i - \sum_{j \neq i} A_{ij} x_j^{(k-1)}\right) / A_{ii}, \quad i = 1, \cdots, n$$

△ Gauss-Seidel iterations:

Initialize $x^{(0)} \in \mathbb{R}^n$, repeat for $k = 1, 2, 3, \cdots$,

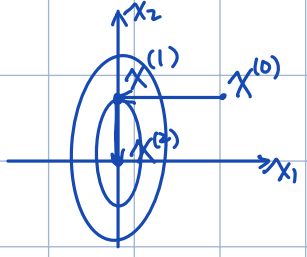
$$x_i^{(k)} = \left(b_i - \sum_{j<i} A_{ij} x_j^{(k)} - \sum_{j>i} A_{ij} x_j^{(k-1)}\right) / A_{ii}, \quad i = 1, \cdots, n$$

· Gauss-Seidel iterations always converge, but Jacobi iterations do not.

Ex. $\min_x f(x)$ $f(x)=4x_1^2+x_2^2$

We use coordinate descent method with exact line search to solve this problem.

It takes 2 iterations

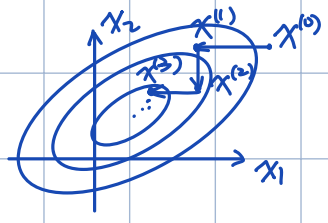


Generally for $f(x)=f_1(x_1)+f_2(x_2)+\cdots+f_n(x_n)$ (Separable) coordinate descent method converges fast

Ex. $\min_x f(x)$ $f(x)=4x_1^2+x_2^2-2x_1x_2$

We use coordinate descent method with exact line search to solve this problem.

It takes more than 2 iterations



~~7~~ Oct. 12th 2021

Conjugate gradient method (Chapter 5 in the textbook)

Consider the problem $\min_x \phi(x)$, $\phi(x)=\frac{1}{2}x^TAx-b^Tx$, $A\succ 0$

Hope: we can solve this problem in n steps.

Def: A set of nonzero vectors $\{p_0,p_1,\cdots,p_k\}$ is said to be conjugate w.r.t. the symmetric positive definite matrix A if

$$p_i^TAp_j=0 \text{ for all } i\neq j$$

Any set of vectors satisfying this property is linearly independent.

We can minimize $\phi(x)$ in n steps by successively minimizing it along the directions in a conjugate set.

Conjugate direction algorithm

(Also true for coordinate descent)

x_0: starting point

$\{p_0, p_1, \cdots, p_n\}$ set of conjugate directions

p_k is not necessarily descent direction, so backtracking line search may fail!

$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$, $r_k = Ax_k - b$ → residual of the linear system

$= -(Ax_k - b)^T p_k / p_k^T A p_k$ → gradient of ϕ at x_k

$x_{k+1} = x_k + \alpha_k p_k$

Theorem: For any $x_0 \in \mathbb{R}^n$, the conjugate direction algorithm converges to the solution x^* in at most n steps. (for the quadratic minimization problem)

Pf. Since the directions $\{p_i\}$ are linear independent, they must span the whole space $\mathbb{R}^n$. Hence we can write the difference between x_0 and the solution x^* as

$$x^* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + \cdots + \sigma_{n-1} p_{n-1}$$

for some choice of scalars σ_k.

Premultiplying this expression by $p_k^T A$ and using the conjugate property, we have

$$p_k^T A(x^* - x_0) = \sigma_k p_k^T A p_k$$

$$\Rightarrow \sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k}$$

Note that $x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \cdots + \alpha_{k-1} p_{k-1}$

Premultiplying this expression by $p_k^T A$ and using the conjugate property, we have $p_k^T A(x_k - x_0) = 0$

Thus $p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k + x_k - x_0)$

$= p_k^T A(x^* - x_k)$

$= p_k^T(b - Ax_k) = -p_k^T r_k$

So $\alpha_k = \alpha_k$

Let $S = \{p_0, p_1, \cdots, p_{n-1}\}$, then S^TAS is diagonal.
We can transform the problem $\min_x \phi(x)$ to $\min_y \phi(Sy)$.
where $y = S^{-1}x$

$$\phi(Sy) = \frac{1}{2}y^T\underbrace{S^TAS}_{\text{diagonal}}y - \underbrace{b^TS}_{(S^Tb)^T}y \equiv \hat{\phi}(y) \left(= \sum_{i=1}^n (c_i y_i^2 + b_i y_i) = \sum_{i=1}^n f_i(y_i)\right)$$

separable

Theorem (Expanding subspace minimization)
Let $x_0 \in \mathbb{R}^n$ be any starting point and suppose that the sequence $\{x_k\}$ is generated by the conjugate direction algorithm. Then

$$r_k^T p_i = 0 \quad \text{for } i = 0, 1, \cdots, k-1$$

and x_k is the minimizer of $\phi(x) = \frac{1}{2}x^TAx - b^Tx$ over the set

$$\{x \mid x = x_0 + \text{span}\{p_0, p_1, \cdots, p_{k-1}\}\}$$

How to find conjugate directions $p_0, p_1, \cdots, p_{n-1}$?

- Eigenvectors of A : $v_1, v_2, \cdots, v_n$ > too expensive computationally $O(n^3)$
- Gram-Schmidt approach

Conjugate gradient method - Preliminary version

$$p_k = -r_k + \beta_k p_{k-1}, \quad p_0 = -\nabla f(x_0) \quad \text{(steepest descent direction at the initial point } x_0\text{)}$$

Since $p_{k-1}^TAp_k = 0$, $\beta_k = \dfrac{r_k^TAp_{k-1}}{p_{k-1}^TAp_{k-1}}$

Conjugate gradient algorithm (preliminary version)

(1) Initialize x_0, $k=0$, $p_0 = -\nabla f(x_0)$, $r_0 = Ax_0 - b$ (for quadratic function)

(2) While $r_k \neq 0$

(a) $\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$

(b) $x_{k+1} = x_k + \alpha_k p_k$

(c) $r_{k+1} = Ax_{k+1} - b$ (for quadratic function)

(d) $\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$

(e) $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$

(f) $k = k+1$

end

Output: $x^* = x_k$, a stationary point of $f(x)$

Why $\{p_0, \cdots, p_{n-1}\}$ is a set of conjugate directions?

Theorem: Suppose that the k'th iterate generated by the conjugate gradient method is not the solution x^*. The following four properties hold:

1° $p_k^T A p_i = 0$, $i = 0, 1, \cdots, k-1$

2° $r_k^T r_i = 0$, $i = 0, 1, \cdots, k-1$

3° $\text{span}\{r_0, r_1, \cdots, r_k\} = \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$

4° $\text{span}\{p_0, p_1, \cdots, p_k\} = \underbrace{\text{span}\{r_0, Ar_0, \cdots, A^k r_0\}}_{\text{krylov subspace of degree } k \text{ for } r_0}$

Therefore, the sequence $\{x_k\}$ converges to x^* in at most n steps

Pf: The proof is by induction

We assume that $\text{span}\{r_0, r_1, \cdots, r_k\} = \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$

$\text{span}\{p_0, p_1, \cdots, p_k\} = \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$

Then $r_k \in \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$

$p_k \in \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}$

$Ap_k \in \text{span}\{Ar_0, A^2 r_0, \cdots, A^{k+1} r_0\}$

Since $r_{k+1} = Ax_{k+1} - b = A(x_{k+1} - x_k) - b + Ax_k$

$= r_k + \alpha_k A p_k$

we have $r_{k+1} \in \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}$

Thus $\text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\} \subset \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}$

Now $A^{k+1} r_0 = A(A^k r_0) \in \text{span}\{Ap_0, Ap_1, \cdots, Ap_k\}$

Since $r_{i+1} = r_i + \alpha_i A p_i$, $Ap_i = \frac{1}{\alpha_i}(r_{i+1} - r_i)$, $\forall i = 0, 1, \cdots, k$, then

$A^{k+1} r_0 \in \text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\}$

Thus $\text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\} \subset \text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\}$

We then have $\text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\} = \text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\}$

Now $\text{span}\{p_0, p_1, \cdots, p_k, p_{k+1}\}$

$= \text{span}\{p_0, p_1, \cdots, p_k, r_{k+1}\}$

$= \text{span}\{r_0, Ar_0, \cdots, A^k r_0, r_{k+1}\}$

$= \text{span}\{r_0, r_1, \cdots, r_k, r_{k+1}\}$

$= \text{span}\{r_0, Ar_0, \cdots, A^{k+1} r_0\}$ (3°, 4° proof finished)

Now $p_1^T A p_0 = 0$ holds from definition

Suppose that it holds for k,

$p_{k+1}^T A p_i = -r_{k+1}^T A p_i + \beta_{k+1} p_k^T A p_i$

If $i = k$, $p_{k+1}^T A p_i = 0$ by construction

If $i \le k-1$, since $p_0, \cdots, p_k$ are conjugate, we have

$r_{k+1}^T p_i = 0$, $i = 0, \cdots, k$ (by last theorem)

Note that $Ap_i \in A \cdot \text{span}\{r_0, Ar_0, \cdots, A^i r_0\} = \text{span}\{Ar_0, A^2 r_0, \cdots, A^{i+1} r_0\}$

$\subset \text{span}\{p_0, p_1, \cdots, p_{i+1}\}$

So $r_{k+1}^T A p_i = 0$ for $i = 0, 1, \cdots, k-1$

Thus $p_{k+1}^T A p_i = -r_{k+1}^T A p_i + \beta_{k+1} p_k^T A p_i = 0$

$\{p_0, \cdots, p_k\}$ is indeed a conjugate direction set, so the algorithm

terminates in at most n iterations (1° proof finished)

Now since $r_k^T p_i = 0$, $i = 0, \cdots, k-1$ and $k = 1, 2, \cdots, n-1$

For $i \geq 1$, $p_i = -r_i + \beta_i p_{i-1}$, then $r_i \in \text{span}\{p_i, p_{i-1}\}$

Thus $r_k^T r_i = 0$ for $i = 1, \cdots, k-1$

Because $r_k^T r_0 = -r_k^T p_0 = 0$, we have $r_k^T r_i = 0$, $i = 0, \cdots, k-1$ (2° proof finished)

Based on this theorem,

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} = -\frac{r_k^T(-r_k + \beta_k r_{k-1})}{p_k^T A p_k} = \frac{r_k^T r_k}{p_k^T A p_k}$$

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \frac{\frac{1}{\alpha_k} r_{k+1}^T (r_{k+1} - r_k)}{\frac{1}{\alpha_k} p_k^T (r_{k+1} - r_k)} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$p_k = -r_k + \beta_{k+1} p_{k-1}$

Fletcher-Reeves method

Conjugate gradient algorithm (standard version)

(1) Initialize x_0, $k = 0$, $p_0 = -\nabla f(x_0)$, $r_0 = Ax_0 - b$ (for quadratic function)

(2) While $r_k \neq 0$

(a) $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$

(b) $x_{k+1} = x_k + \alpha_k p_k$

(c) $r_{k+1} = r_k + \alpha_k A p_k$

(d) $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$

(e) $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$

(f) $k = k + 1$

end

Output $x^* = x_k$, a stationary point of $f(x)$

For conjugate gradient method, we only need to store x, r and

p in the last iteration. It is recommended for large problems.

7: Oct. 14th 2021

Rate of convergence

Theorem. If A has only r distinct eigenvalues, then the conjugate gradient iterations will terminate at the solution in at most r iterations.

Theorem. If A has eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, we have that

$$\|x_{k+1} - x^*\|_A^2 \le \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|x_0 - x^*\|_A^2$$

$\|x\|_A^2 = x^T A x$

$$k=0: \quad \|x_1 - x^*\|_A^2 \le \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|x_0 - x^*\|_A^2$$

Nonlinear conjugate gradient method

Fletcher-Reeves Method

(1) Initialize x_0, $p_0 = -\nabla f(x_0)$, $k = 0$

(2) While $\nabla f_k \neq 0$

(a) Find α_k such that α_k satisfy the strong Wolfe conditions with $0 < c_2 < \frac{1}{2}$

(b) $x_{k+1} = x_k + \alpha_k p_k$

(c) Find ∇f_{k+1}

(d) $\beta_{k+1}^{FR} = \dfrac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$

(e) $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k$

(f) $k = k+1$

end

Output: $x^* = x_k$, a stationary point of $f(x)$

Lemma. For FR algorithm, strong Wolfe conditions guarantees descent direction.

Assumption: · The level set $\mathcal{L}=\{x: f(x)\le f(x_0)\}$ is bounded

· $f \in C^2$

· α_k satisfies strong Wolfe conditions

$$f(x_k+\alpha_k p_k) \le f(x_k)+c_1\alpha_k \nabla f_k^T p_k$$

$$|\nabla f(x_k+\alpha_k p_k)^T p_k| \le -c_2 \nabla f_k^T p_k, \text{ with } 0<c_1<c_2<\tfrac{1}{2}$$

Then $-\frac{1}{1-c_2} \le \frac{\nabla f_k^T p_k}{\|\nabla f_k\|^2} \le \frac{2c_2-1}{1-c_2}$ for all $k=0,1,\dots$

7=Oct. 19th. 2021

Polak-Ribière method

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T(\nabla f_{k+1}-\nabla f_k)}{\nabla f_k^T \nabla f_k}$$

We have $\beta_{k+1}^{PR} = \beta_{k+1}^{FR}$ for strongly convex quadratic function and exact line search. When applied to the nonlinear functions with inexact line search, PR and FR differs markedly. Numerical experiments indicate that PR tends to be more robust and efficient of the two.
However, strong Wolfe conditions do not guarantee that p_k is always a descent direction in PR.

Polak Ribière + method

$$\beta_{k+1}^{PR+} = \max\{\beta_{k+1}^{PR}, 0\}$$

The strong Wolfe conditions ensures that the descent property

holds for PR+

Hestenes-Stiefel formula

$$\beta_{k+1}^{HS} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k}$$

Similar to PR, both in terms of theoretical convergence properties and in its

△ Restart

Restart the iteration at every n steps by setting $\beta_k = 0$, i.e., by taking a steepest descent step. Theoretically, it leads to n-step quadratic convergence, i.e., $\|x_{k+n} - x^*\| = O(\|x_k - x^*\|^2)$.
Finite termination will occur within n steps of the restart if the function can be approximated by a quadratic form.
There are other criterion for restart for large problems, such that

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\|\nabla f_k\|^2} \geq \nu \quad , \text{a typical value for } \nu \text{ is } 0.1$$

Rational: the gradients are mutually orthogonal when f is a quadratic function.
Powell (1977) "Restart procedure for the conjugate gradient method":

$$-1.2\|\nabla f_k\|^2 \leq p_k^T \nabla f_k \leq -0.8\|\nabla f_k\|^2$$

Global convergence

Assumptions:

(i) The level set $\mathcal{L} = \{x \mid f(x) \leq f(x_0)\}$ is bounded

(ii) In some open neighborhood $\mathcal{N}$ of $\mathcal{L}$, the objective function f is Lipschitz continuously differentiable

Theorem: Suppose that the assumptions hold, and that the Fletcher-Reeves method is implemented with a line search that satisfies the strong Wolfe conditions with $0<c_1<c_2<\frac{1}{2}$. Then $\liminf_{k\to\infty} \|\nabla f_k\| = 0$

Theorem: Consider the Polak-Ribière method with an ideal line search. There exists a C^2 objective function $f:\mathbb{R}^3\to\mathbb{R}$ and a starting point $x_0\in\mathbb{R}^3$ such that the sequence of gradients $\{\|\nabla f_k\|\}$ is bounded away from zero.

"Ideal line search" means that the line search returns a value α_k that is the first positive stationary point for the function $t(\alpha)=f(x_k+\alpha p_k)$

Remark: $\lim_{k\to\infty}\|\nabla f_k\|=0$ can be established for the Polak-Ribière method under the assumption that f is strongly convex and an exact line search is used.

Derivative-free optimization

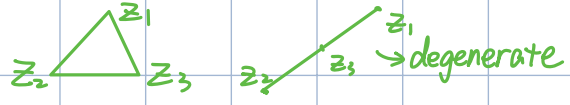
Nelder-Mead method (Sec 9.5 of the textbook)

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

Given a simplex S with vertices $\{z_1, z_2, \cdots, z_{n+1}\}$, define a matrix

$V(S) = [z_2 - z_1, z_3 - z_1, \cdots, z_{n+1} - z_1]$

(edge along v from z_1)



This simplex is said to be nondegenerate or nonsingular if V is a nonsingular matrix.

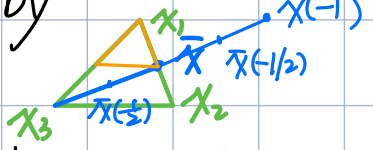
In a single iteration of the Nelder-Mead algorithm, we seek to remove the vertex with the worst function value and replace it with another point with a better value. The new point is obtained by reflecting, expanding, or contracting the simplex along the line joining the worst vertex with the centroid of the remaining vertices.

If we cannot find a better point in this manner, we retain only the vertex with the best function value, and we shrink the simplex by moving all other vertices toward this value.

Suppose that the $n+1$ vertices of the current simplex are denoted by $\{x_1, x_2, \cdots, x_{n+1}\}$ where $f(x_1) \le f(x_2) \le \cdots \le f(x_{n+1})$

The centroid of the best n points is denoted by

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$



Points along the line joining $\bar{x}$ and the "worst" vertex x_{n+1} are denoted by $\bar{x}(t) = \bar{x} + t(x_{n+1} - \bar{x})$

One step of Nelder-Mead Simplex

Compute the reflection point $\bar{x}(-1)$ and evaluate $f_{-1} = f(\bar{x}(-1))$

△ If $f(x_1) \le f_{-1} < f(x_n)$, replace x_{n+1} by $\bar{x}(-1)$ and go to next iteration.

(reflected point is neither best nor worst in the new simplex)

else if $f_{-1} < f(x_1)$,

compute the expansion point $\bar{x}(-2)$ and evaluate $f_{-2} = f(\bar{x}(-2))$

if $f_{-2} < f_{-1}$,

replace x_{n+1} by $\bar{x}(-2)$ and go to next iteration

else,

replace x_{n+1} by $\bar{x}(-1)$ and go to next iteration

else if $f_{-1} \geq f(x_n)$, (reflected point is still worse than x_n)

if $f(x_n) \leq f_{-1} < f(x_{n+1})$,

evaluate $f_{-1/2} = \bar{x}(-1/2)$. (perform outside contraction)

if $f_{-1/2} \leq f_{-1}$,

replace x_{n+1} by $\bar{x}(-\frac{1}{2})$ and go to next iteration

else, go to next else

else,

evaluate $f_{1/2} = \bar{x}(1/2)$ (perform inside contraction)

if $f_{1/2} < f_{n+1}$,

replace x_{n+1} by $\bar{x}(\frac{1}{2})$ and go to next iteration

replace $x_i \leftarrow \frac{1}{2}(x_1 + x_i)$ for $i = 2, 3, \cdots, n+1$

(neither outside nor inside contraction was acceptable, shrink the simplex toward x_1)

※ Oct. 21st 2021

Subgradients

If you can compute subgradients, then you can minimize any convex function.

Recall that for convex and differentiable f,

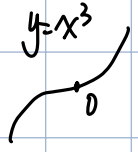
$f(y) \geq f(x) + \nabla f(x)^T (y-x)$ for all x, y

This is a linear approximation always underestimates f.

A subgradient of a convex function f at x is any $g \in \mathbb{R}^n$ such that

$f(y) \geq f(x) + g^T(y-x)$ for all y.

- Always exists for convex function (on the relative interior of the the domain of f)
- May not exist for nonconvex functions



- If f is differentiable at x, then $g = \nabla f$ uniquely.

Examples:

1. $f: \mathbb{R} \to \mathbb{R}$, $f(x) = |x|$

   For $x \neq 0$, unique subgradient $g = \text{sign}(x)$

   For $x = 0$, g is any element of $[-1, 1]$.

2. $f: \mathbb{R}^n \to \mathbb{R}$, $f(x) = \|x\|_2 = \sqrt{x^T x}$

   For $x \neq 0$, unique subgradient $g = \frac{x}{\|x\|_2}$

   For $x = 0$, g is any element of $\{z : \|z\| \leq 1\}$

   check: $f(y) = \|y\|_2 \geq \|y\|_2 \|g\|_2 \geq |g^T y| \geq g^T y$

3. $f: \mathbb{R}^n \to \mathbb{R}$ $f(x) = \|x\|_1 = \sum_{i=1}^n |x_i|$

   For $x \neq 0$, unique subgradient $g = \begin{pmatrix} \text{sign}(x_1) \\ \vdots \\ \text{sign}(x_n) \end{pmatrix}$

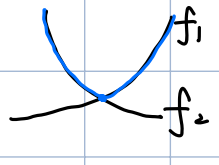
   For $x = 0$, ith component g_i is any element of $[-1, 1]$.

4. $f = \max\{f_1(x), f_2(x)\}$ for $f_1, f_2: \mathbb{R}^n \to \mathbb{R}$ convex, differentiable

   For $f_1(x) > f_2(x)$, unique subgradient $g = \nabla f_1(x)$

   For $f_1(x) < f_2(x)$, unique subgradient $g = \nabla f_2(x)$

   For $f_1(x) = f_2(x)$, g is any point on the line segment between $\nabla f_1(x)$ and $\nabla f_2(x)$, i.e., $\alpha \nabla f_1(x) + (1-\alpha) \nabla f_2(x)$; $\alpha \in [0, 1]$



Subdifferential

Set of all subgradients of convex function f is called the subdifferential

$$\partial f(x) = \{g \in \mathbb{R}^n : g \text{ is a subgradient of } f \text{ at } x\}$$

- Nonempty
- $\partial f(x)$ is closed and convex (True even for nonconvex f)
- If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$
- If $\partial f = \{g\}$, then f is differentiable at x and $\nabla f(x) = g$

Basic rules for subdifferential (convex functions)

- Scaling $\partial(af) = a\partial f$ provided $a > 0$
- Addition $\partial(f_1 + f_2) = \partial f_1 + \partial f_2$
- Affine composition If $g(x) = f(Ax+b)$, then $\partial g(x) = A^T \partial f(Ax+b)$
- Finite pointwise maximum

  If $f(x) = \max_{i \in [m]} f_i(x)$, then $\partial f(x) = \text{conv}\left(\bigcup_{i: f_i(x) = f(x)} \partial f_i(x)\right)$

  convex hull of union of subdifferentials of active functions at x.
- General pointwise maximum

  If $f(x) = \max_{s \in S} f_s(x)$, then $\partial f(x) \supseteq \text{closure}\{\text{conv}\left(\bigcup_{s: f_s(x) = f(x)} \partial f_s(x)\right)\}$

  Under some regularity conditions (on S, f_s) we get equality.
- Norms $f(x) = \|x\|_p$ $(p \geq 1)$

  Let q be such that $\frac{1}{p} + \frac{1}{q} = 1$, then $\|x\|_p = \max_{\|z\|_q \leq 1} z^T x$

  And $\partial f(x) = \underset{\|z\|_q \leq 1}{\text{argmax}}\ z^T x$

※ Oct. 26th. 2021

Optimality condition

For any f, $f(x^*) = \min_x f(x) \iff 0 \in \partial f(x^*)$

i.e., x^* is a minimizer if and only if 0 is a subgradient of f at x^*

This is called the subgradient optimality condition.

pf: $0 \in \partial f(x^*) \Leftrightarrow f(y) \geq f(x^*) + 0(y - x^*) = f(x^*)$

Example: Lasso optimality conditions

Given $Y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, lasso problem can be parameterized as

$$\min_\beta \frac{1}{2}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1, \text{ where } \lambda \geq 0$$

Subgradient optimality

$$0 \in \partial\left(\frac{1}{2}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1\right)$$

$$\Leftrightarrow 0 \in -X^T(Y - X\beta) + \lambda \partial\|\beta\|_1$$

$$\Leftrightarrow X^T(Y - X\beta) = \lambda v$$

for some $v \in \partial\|\beta\|_1$, i.e., $v_i = \begin{cases} \{1\} & \text{if } \beta_i > 0 \\ \{-1\} & \text{if } \beta_i < 0 \\ [-1, 1] & \text{if } \beta_i = 0 \end{cases}$, $i = 1, \dots, p$

Write $X_1, \dots, X_p$ for columns of X. Then our condition reads

$$\begin{cases} X_i^T(Y - X\beta) = \lambda \cdot \text{sign}(\beta_i) & \text{if } \beta_i \neq 0 \\ |X_i^T(Y - X\beta)| \leq \lambda & \text{if } \beta_i = 0 \end{cases}$$

Note: Subgradient optimality condition don't lead to closed-form expression for a lasso problem. However, they do provide a way to check lasso optimality (used by screening rules)

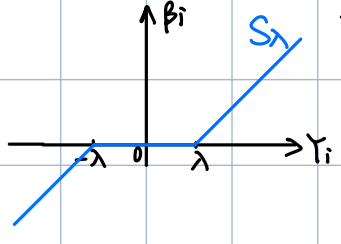
Example (cont'd) Simplified lasso problem with $X = I$

$$\min_\beta \frac{1}{2}\|Y - \beta\|_2^2 + \lambda\|\beta\|_1$$

This time we can solve directly using subgradient optimality.

Solution is $\beta = S_\lambda(Y)$, where S_λ is the soft-thresholding operator

$$\beta_i = [S_\lambda(Y)]_i = \begin{cases} Y_i - \lambda & \text{if } Y_i > \lambda \\ 0 & \text{if } -\lambda \leq Y_i \leq \lambda \\ Y_i + \lambda & \text{if } Y_i < -\lambda \end{cases}$$



Check subgradient optimality conditions

if $\beta \neq 0$, $Y_i - \beta_i = \lambda = \lambda \cdot \text{sign}(\beta_i)$ if $\beta_i > 0$

$Y_i - \beta_i = -\lambda = \lambda \cdot \text{sign}(\beta_i)$ if $\beta_i < 0$

if $\beta = 0$, $|Y_i - \beta_i| = |Y_i - 0| = |Y_i| \leq \lambda$

So $\beta = S_\lambda(Y)$ satisfies the subgradient optimality condition.

Example: Distance to a convex set

$\text{dist}(x, C) = \min_{y \in C} \|y - x\|_2$, $x \notin C$, C convex set

This is a convex function

Write $\text{dist}(x, C) = \|x - P_C(x)\|_2$, where $P_C(x)$ is the projection of x onto C. It turns out that when $\text{dist}(x, C) > 0$, $\partial \text{dist}(x, C) = \left\{ \frac{x - P_C(x)}{\|x - P_C(x)\|} \right\}$

Since $\partial \text{dist}(x, C)$ only has one element, so in fact $\text{dist}(x, C)$ is differentiable and this is its gradient.

Subgradient method

Steepest descent method considers the problem $\min_{x \in \mathbb{R}^n} f(x)$, where $f \in D$.

After choosing an initial value $x_0 \in \mathbb{R}^n$. repeat $x_{k+1} = x_k - \alpha_k \nabla f_k$, $k = 0, 1, 2, \cdots$

Subgradient method considers the problem $\min_{x \in \mathbb{R}^n} f(x)$, where f is convex.

After choosing an initial value $x_0 \in \mathbb{R}^n$. repeat $x_{k+1} = x_k - \alpha_k g_k$, $g_k \in \partial f(x_k)$, i.e., any subgradient of f at x_k.

*Subgradient method is not necessarily a descent method, so we keep track of the best iterate x_k, best among $x_0, \cdots, x_k$ so far, i.e., $f(x_{k,\text{best}}) = \min_{i=0,\cdots,k} f(x_i)$

Step size choices

- Fixed step size: $\alpha_k = \alpha$ for all $k = 0, 1, \cdots$
- Diminishing step size: Choose to meet conditions

$$\sum_{k=0}^{+\infty} \alpha_k^2 < \infty \quad , \quad \sum_{k=0}^{+\infty} \alpha_k = \infty$$

i.e., square summable but not summable

It is different from the steepest descent method that the step sizes are pre-specified, not adaptively computed.

Convergence analysis

Assume that f is convex, $dom(f) = \mathbb{R}^n$, and also that f is Lipschitz continuous with $G > 0$, i.e.,

$$|f(x) - f(y)| \leq G\|x - y\|_2 \quad \text{for all } x, y$$

Theorem: For a fixed step size α, subgradient method satisfies

$$\lim_{k \to \infty} f(x_{k,best}) \leq f(x^*) + \frac{G^2 \alpha}{2}$$

Theorem: For diminishing step sizes, subgradient method satisfies

$$\lim_{k \to \infty} f(x_{k,best}) = f(x^*)$$

Pf: $\|x_{k+1} - x^*\|^2 = \|x_k - \alpha_k g_k - x^*\|^2$

$$= \|x_k - x^*\|^2 - 2\alpha_k g_k^T (x - x^*) + \alpha_k^2 \|g_k\|^2$$

By the definition of the subgradient method, we have

$$f(x^*) \geq f(x_k) + g_k^T (x^* - x_k)$$

$$\Rightarrow -g_k^T (x_k - x^*) \leq -[f(x_k) - f(x^*)]$$

Using this inequality, we have

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k [f(x_k) - f(x^*)] + \alpha_k^2 \|g_k\|^2$$

$$\leq \|x_0 - x^*\|^2 - 2 \sum_{i=0}^{k} \alpha_k [f(x_i) - f(x^*)] + \sum_{i=0}^{k} \alpha_i^2 \|g_i\|^2$$

The left hand side is lower bounded by 0, then we have

$$0 \le \|x_0 - x^*\|^2 - 2\sum_{i=0}^{k}\alpha_i[f(x_i) - f(x^*)] + \sum_{i=0}^{k}\alpha_i^2\|g_i\|^2$$

$$\Rightarrow 2\sum_{i=0}^{k}\alpha_i[f(x_i) - f(x^*)] \le \|x_0 - x^*\|^2 + \sum_{i=0}^{k}\alpha_i^2\|g_i\|^2$$

$$\Rightarrow f(x_{k,best}) - f(x^*) \le \frac{\|x_0 - x^*\|^2 + \sum_{i=0}^{k}\alpha_i^2\|g_i\|^2}{2\sum_{i=0}^{k}\alpha_i} \triangleq \text{RHS}$$

f Lipschitz $\Rightarrow \|g\| \le G$, $\forall g \in \partial f$

For a constant step size $\alpha_i = \alpha$, $\text{RHS} \le \frac{\|x_0 - x^*\|^2 + k\alpha^2G^2}{2\alpha k} \to \frac{G^2\alpha}{2}$

For diminishing step sizes with $\sum_{i=1}^{\infty}\alpha_i^2 < \infty$, $\sum_{i=0}^{\infty}\alpha_i = \infty$,

$$\text{RHS} \le \frac{\|x_0 - x^*\|^2 + G\sum_{i=0}^{k}\alpha_i^2}{2\sum_{i=1}^{k}\alpha_i} \to 0 \Rightarrow \lim_{k\to\infty} f(x_{k,best}) = f(x^*)$$

$$f(x_{k,best}) - f(x^*) \le \frac{\|x_0 - x^*\|^2}{2k\alpha} + \frac{G^2\alpha}{2}$$

$< \varepsilon/2$ $\quad$ $< \varepsilon/2$

$k > \frac{\|x_0 - x^*\|^2}{\alpha\varepsilon}$ $\quad$ $\alpha < \varepsilon/G^2$

$\gtrsim G^2\|x_0 - x^*\|^2/\varepsilon^2 \Rightarrow k = O(\frac{1}{\varepsilon^2})$

steepest descent: $k = O(\frac{1}{\varepsilon})$

To make $f(x_{k,best}) - f(x^*) \le \varepsilon$, the subgradient method needs $O(\frac{1}{\varepsilon^2})$ iterations. Note that steepest descent method needs $O(\frac{1}{\varepsilon})$.

Polyak step size

When the optimal value $f(x^*)$ is known, take

$$\alpha_k = \frac{f(x_k) - f(x^*)}{\|g_k\|_2^2}, \quad k = 0, 1, 2, \cdots$$

It is motivated from an inequality in the subgradient convergence proof: $\|x_{k+1} - x^*\|^2 \le \|x_k - x^*\|^2 - 2\alpha_k[f(x_k) - f(x^*)] + \alpha_k^2\|g_k\|_2^2$

Polyak step size minimize the right hand side.

With Polyak step sizes, the subgradient method converges to the optimal value, and the convergence rate is still $O(\frac{1}{\varepsilon^2})$.

Example: Intersection of sets.

Suppose we want to find $x^* \in C_1 \cap \cdots \cap C_m$, i.e., find a point

in the intersection of closed, convex set $C_1, \cdots, C_m$.

First define $f_i(x) = \text{dist}(x, C_i)$, $i = 1, \cdots, m$

$$f(x) = \max_{i \in [m]} f_i(x)$$

and now solve $\min_x f(x)$.

Note that $f(x)$ is convex

$f(x^*) = 0 \iff x^* \in C_1 \cap \cdots \cap C_m$

Recall the distance function $\text{dist}(x, C) = \min_{y \in C} \|y - x\|_2$, and

$\nabla \text{dist}(x, C) = \frac{x - P_C(x)}{\|x - P_C(x)\|_2}$

Also recall that if $f(x) = \max_{i \in [m]} f_i(x)$, then $\partial f(x) = \text{conv}(\bigcup_{i: f_i(x) = f(x)} \partial f_i(x))$

So if $f_i(x) = f(x)$ and $g_i \in \partial f_i(x)$, then $g_i \in \partial f(x)$.

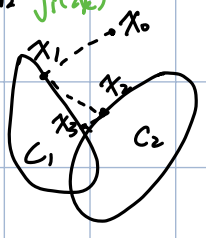
Put these two facts together, if C_i is farthest set from x, and $g_i = \nabla f_i(x) = \frac{x - P_{C_i}(x)}{\|x - P_{C_i}(x)\|_2}$, then $g_i \in \partial f(x)$.

Now apply subgradient method with Polyak step size α_k

Suppose C_i is farthest from x_k, we have

$$x_{k+1} = x_k - \alpha_k g_k = x_k - \frac{f(x_k) - f(x^*)}{\|g_k\|_2^2} \cdot \frac{x_k - P_{C_i}(x_k)}{\|x_k - P_{C_i}(x_k)\|_2}$$

$$= x_k - (x_k - P_{C_i}(x_k)) = P_{C_i}(x_k)$$



For two sets, this is the famous alternating projection algorithm.

(von Neumann, 1950. "Functional operators, volume II: The geometry of orthogonal space), i.e, just keep projecting back and forth between the sets.

To optimize a convex function f over a convex set C,

$$\min_x f(x) \quad \text{subject to } x \in C$$

we can use the <u>projected subgradient method</u>. Just like the usual subgradient method, except we project onto C at each

iterations $x^{k+1} = P_C(x^k - \alpha_k g^k)$, $k = 0, 1, 2, \cdots$

Assuming we can do this projection, we get the same convergence guarantees as the usual subgradient method, with the same step size choices.

Note: projected gradient descent works too.

What sets C are easy to project onto?

- Affine images $\{Ax + b : x \in \mathbb{R}^n\}$
- Solution set of linear system $\{x : Ax = b\}$
- Nonnegative orthant: $\mathbb{R}^n_+ = \{x : x \geq 0\}$
- Some norm balls $\{x : \|x\|_p \leq 1\}$ for $p = 1, 2, \infty$
- Some simple polyhedra and simple cones

Warning: it is easy to write down seemingly simple set C, and P_C can turn out to be very hard. For example, generally hard to project onto arbitrary polyhedron $C = \{x : Ax \leq b\}$. ~~$A_1 x \leq b_1$, $A_2 x \leq b_2$~~

Oct. 28th. 2021

Example: Regularized regression methods

Given $(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}$ for $i = 1, \cdots, n$, the least square loss is

$$f(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2, \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad X = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix}$$

Consider the regularized problem

$$\min_\beta f(\beta) + \lambda P(\beta) \quad \text{where} \quad P(\beta) = \|\beta\|_2^2 \text{ ridge penalty}$$

$$P(\beta) = \|\beta\|_1, \text{ lasso penalty}$$

Use steepest descent method for ridge regression

Use subgradient method for lasso regression

△ For subgradient method, the upside is its broad application and

the downside is the slow convergence rate over problems of convex, Lipschitz function.

Theorem (Nesterov): For nonsmooth first-order iterative method that updates x_{k+1} in $x_0 + \text{span}\{g_0, g_1, \cdots, g_k\}$, where subgradients $g_0, \cdots, g_k$ comes from weak oracle, if $k \le n-1$, there is a function in the problem class such that any nonsmooth first-order method satisfies: $f(x_k) - f(x^*) \ge \frac{\|x_0 - x^*\| G}{2(1+\sqrt{k+1})}$ ← Lipschitz constant

Nov. 2nd 2021

Proximal gradient descent

Also called composite gradient descent or generalized gradient descent (Beck and Teboulle (2008) "A fast iterative shrinkage-thresholding algorithm for linear inverse problems")

Consider decomposable functions

$$f(x) = g(x) + h(x)$$

where g is convex, differentiable, $\text{dom}(g) = \mathbb{R}^n$

h is convex, not necessarily differentiable

If f were differentiable, then steepest descent update would be

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

The steepest descent method can be motivated by minimizing quadratic approximation to f around x_k, replace $\nabla^2 f(x_k)$ by $\frac{1}{\alpha_k} I_n$

$$x_{k+1} = \arg\min_x f(x) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2\alpha_k} \|x - x_k\|_2^2$$

Now in our case, f is not differentiable, but $f = g + h$, $g \in D$

We can make quadratic approximation to g and leave h alone, i.e., update $x_{k+1} = \arg\min_x g(x_k) + \nabla g(x_k)^T (x - x_k) + \frac{1}{2\alpha_k} \|x - x_k\|_2^2 + h(x)$

$$= \arg\min_x \frac{1}{2\alpha_k}\|x-(x_k-\alpha_k\nabla g(x_k))\|_2^2 + h(x)$$

stay close to steepest descent update for g ; also make h small

Define proximal mapping

$$\text{prox}_\alpha(x) = \arg\min_z \frac{1}{2\alpha}\|z-x\|^2 + h(z)$$

Proximal gradient descent:

Choose initial value x_0, then repeat

$$x_{k+1} = \text{prox}_{\alpha_k}(x_k - \alpha_k \nabla g(x_k)), \quad k=0,1,2,\cdots$$

To make this update look familiar to the steepest descent, we can write it as $x_{k+1} = x_k - \alpha_k G_{\alpha_k}(x_k)$

where G_α is the generalized gradient of f

$$G_\alpha(x) = \frac{x - \text{prox}_\alpha(x-\alpha\nabla f(x))}{\alpha}$$

Note: Mapping $\text{prox}_\alpha(\cdot)$ does not depend on g at all, only on h

- Smooth part g can be complicated, we only need to compute its gradient
- Each iteration evaluates $\text{prox}_\alpha(\cdot)$ once, and this can be cheap or expensive, depending on h.

Example: ISTA (iterative soft thresholding algorithm)

Given $Y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n\times p}$, recall lasso criterion

$$f(\beta) = \underbrace{\frac{1}{2n}\|Y-X\beta\|_2^2}_{g(\beta)} + \underbrace{\lambda\|\beta\|_1}_{h(\beta)}$$

Proximal mapping is now

$$\text{prox}_\alpha(\beta) = \arg\min_x \frac{1}{2\alpha}\|\beta-x\|_2^2 + \lambda\|x\|_1$$
$$= S_{\alpha\lambda}(\beta)$$

where $S_{\alpha\lambda}(\beta)$ is the soft-thresholding operator

$$[S_\lambda(\beta)]_i = \begin{cases} \beta_i-\lambda, & \text{if } \beta_i > \lambda \\ 0, & \text{if } -\lambda \le \beta_i \le \lambda \\ \beta_i+\lambda, & \text{if } \beta_i < -\lambda \end{cases}, \quad i=1,\cdots,n$$

Recall $\nabla g(\beta) = -\frac{1}{n} X^T(Y - X\beta)$, hence the proximal gradient update

$$\beta^{k+1} = S_{\alpha\lambda}(\beta^k + \frac{\alpha^k}{n} X^T(Y - X\beta^k))$$

often called the iterative soft-thresholding algorithm (ISTA)

Backtracking line search for proximal gradient descent works similarly as in steepest descent, but operates on g and not f. Choose parameter $0 < \beta < 1$. At each iteration, start at $\alpha = \alpha_{init}$, and while

$$g(x - \alpha G_\alpha(x)) > g(x) - \alpha \nabla g(x)^T G_\alpha(x) + \frac{\alpha}{2} \|G_\alpha(x)\|_2^2,$$

shrink $\alpha = \beta\alpha$.

Convergence analysis

Assume that $f(x) = g(x) + h(x)$, g is convex, differentiable, $\text{dom}(g) = \mathbb{R}^n$, and ∇g is Lipschitz continuous with constant $L > 0$, h is convex, $\text{prox}_\alpha(x) = \arg\min_z \{\frac{1}{2\alpha}\|x - z\|^2 + h(z)\}$ can be evaluated

Theorem. Proximal gradient descent with fixed step size $\alpha \leq \frac{1}{L}$ satisfies $f(x^k) - f(x^*) \leq \frac{\|x^0 - x^*\|^2}{2\alpha k}$,

and the same result holds for backtracking, with α replaced by $\frac{\beta}{L}$.

Example: Matrix completion

Mazumder et al. (2011) "Spectral regularization algorithms for learning large incomplete matrices"

Given a matrix $Y \in \mathbb{R}^{m \times n}$, and only observe entries $y_{ij}, (i,j) \in \Omega$

Suppose we want to fill in missing entries (e.g. for a recommender system), so we solve a matrix completion problem

$$\min_B \frac{1}{2}\sum_{(i,j)\in\Omega}(y_{ij}-B_{ij})^2 + \lambda\|B\|_{tr}$$

Here $\|B\|_{tr}$ is the trace (or nuclear) norm of B

$$\|B\|_{tr} = \sum_{i=1}^{r}\sigma_i(B)$$

where $r = \text{rank}(B)$ and $\sigma_1(X) \geq \cdots \geq \sigma_r(X) \geq 0$ are the singular values.

Define P_Ω, projection operator onto observed set

$$[P_\Omega(B)]_{ij} = \begin{cases} B_{ij} & (i,j)\in\Omega \\ 0 & (i,j)\notin\Omega \end{cases}$$

Then the criterion is

$$f(B) = \underbrace{\frac{1}{2}\|P_\Omega(Y) - P_\Omega(B)\|_F^2}_{g(B)} + \underbrace{\lambda\|B\|_{tr}}_{h(B)}$$

$\|A\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2$

Two ingradients needed for proximal gradient descent

- Gradient calculation $\nabla g(B) = -(P_\Omega(Y) - P_\Omega(B))$
- Proximal mapping $\text{prox}_\alpha(B) = \underset{Z}{\text{argmin}} \frac{1}{2\alpha}\|B-Z\|_F^2 + \lambda\|Z\|_{tr}$

Fact: $\text{prox}_\alpha(B) = S_{\lambda\alpha}(B)$ matrix soft-thresholding at the level $\alpha\lambda$

Here $S_\lambda(B)$ is define by

$$S_\lambda(B) = U\Sigma_\lambda V^T$$

where $B = U\Sigma V^T$ is an SVD, and Σ_λ is a diagonal matrix with $(\Sigma_\lambda)_{ii} = \max\{\Sigma_{ii} - \lambda, 0\}$

Hence proximal gradient update step is

$$B_{k+1} = S_{\lambda\alpha}(B_k + \alpha(P_\Omega(Y) - P_\Omega(B)))$$

Note that $\nabla g(B)$ is Lipschitz continuous with $L=1$, so we can choose fixed step size $\alpha = 1$. Update step is now

$$B_{k+1} = S_{\lambda\alpha}(P_\Omega(Y) + P_{\Omega^\perp}(B))$$

where $P_{\Omega^\perp}$ projects to the complement of Ω. $P_\Omega(B) + P_{\Omega^\perp}(B) = B$

This is called the soft-impute algorithm, simple and effective

method for matrix completion.

Several special cases

- $h=0$ steepest descent
- $h=I_C$ projected gradient descent
- $g=0$ proximal minimization algorithm

These algorithms all have $O(\frac{1}{\varepsilon})$ convergence rate

△Projected gradient descent

Given closed, convex set $C \in \mathbb{R}^n$,

$$\min_{x \in C} g(x) \iff \min_x g(x) + I_C(x)$$

where $I_C(x) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{if } x \notin C \end{cases}$ is the indicator function of C.

Hence $\text{prox}_\alpha(x) = \arg\min_z \frac{1}{2\alpha}\|x-z\|_2^2 + I_C(x)$

$$= \arg\min_{z \in C} \frac{1}{2\alpha}\|x-z\|_2^2$$

Therefore proximal gradient update step is

$$x_{k+1} = P_C(x_k - \alpha \nabla g(x_k))$$

Perform usual gradient update and then project back onto C

△Proximal minimization algorithm considers h as convex (not necessarily differentiable) function. The optimization problem is $\min_x h(x)$

Proximal gradient update step is just

$$x_{k+1} = \arg\min_z \frac{1}{2\alpha}\|x-z\|_2^2 + h(z)$$

This algorithm is called proximal minimization algorithm.

Faster than subgradient method, but not implementable unless we know the proximal mapping in closed form.

△What if we cannot evaluate the proximal mapping? In general, it is not clear what happens if we just minimizing this approximately. But if you can precisely control the errors in approximating the

proximal mapping, then you can recover the original convergence rate.

(Schmidt et al. (2011) "Convergence rates of inexact proximal gradient method for convex optimization")

In practice, if proximal mapping evaluation is done approximately, then it should be done to decent high accuracy.

Turns out we can accelerate proximal gradient descent in order to achieve the $O(\frac{1}{\sqrt{\varepsilon}})$ convergence rate. Four ideas (three acceleration methods) by Nesterov.

- 1983: original acceleration idea for smooth function
- 1988: another acceleration idea for smooth function
- 2005: Smoothing techniques for nonsmooth functions, coupled with original acceleration idea
- 2007: acceleration idea for composite functions.

We will follow Beck and Teboulle (2008), an extension of Nesterov (1983) to composite functions.

Consider $\min_{x \in \mathbb{R}^n} g(x) + h(x)$

g is convex, differentiable, and h is convex

Accelerated proximal gradient method:

Choose initial point $x_0 = x_{-1} \in \mathbb{R}^n$, repeat for $k = 0, 1, 2, \cdots$

$$v = x_k + \frac{k-2}{k+1}(x_k - x_{k-1})$$

$$x_{k+1} = \text{prox}_{\alpha_k}(v - \alpha_k \nabla g(v))$$

※ Nov. 4th 2021

Accelerated proximal gradient method

Choose initial point $x_0 = x_{-1} \in \mathbb{R}^n$, repeat

$$v = x_{k-1} + \frac{k-2}{k+1}(x_{k-1} - x_{k-2})$$

$$x_k = \text{prox}_{\alpha_{k-1}}(v - \alpha_{k-1}\nabla g(v)), \quad k = 1, 2, 3, \ldots$$

- First two steps are just the usual proximal gradient update
- After that $v = x_{k-1} + \frac{k-2}{k+1}(x_{k-1} - x_{k-2})$ carries some "momentum" from previous iterations
- $h = 0$ gives accelerated gradient method

FISTA

Back to the lasso problem $\min_\beta \frac{1}{2n}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1$

Recall ISTA, $\beta_k = S_{\lambda\alpha_{k-1}}(\beta_{k-1} + \frac{\alpha_{k-1}}{n}X^T(Y - X\beta_{k-1}))$, $k = 1, 2, \ldots$

$S_{\lambda\alpha_{k-1}}(\cdot)$ being vector soft-thresholding operator

Applying acceleration gives us FISTA

$$v = \beta_{k-1} + \frac{k-2}{k+1}(\beta_{k-1} - \beta_{k-2})$$

$$\beta_k = S_{\lambda\alpha_{k-1}}(v + \frac{\alpha_{k-1}}{n}X^T(Y - Xv)), \quad k = 1, 2, \ldots$$

Backtracking under acceleration can be performed in different ways. Simple approach: Fix $\beta < 1$, $\alpha_0 = 1$. At iteration k, start with $\alpha = \alpha_{k-1}$ and let $x^+ = \text{prox}_\alpha(v - \alpha\nabla g(v))$ if

$$g(x^+) > g(v) + \nabla g(v)^T(x^+ - v) + \frac{1}{2\alpha}\|x^+ - v\|_2^2$$

shrink $\alpha = \beta\alpha$, else set $x_k = x^+$.

Note that this strategy forces us to take decreasing step size

Convergence Analysis

Assume that $f(x) = g(x) + h(x)$, g is convex, differentiable, $\text{dom}(g) = \mathbb{R}^n$, and ∇g is Lipschitz continuous with constant $L > 0$, h is convex, $\text{prox}_\alpha(x) = \text{argmin}_z \{\frac{1}{2\alpha}\|x - z\|_2^2 + h(z)\}$ can be evaluated

Theorem. Accelerated proximal gradient method with fixed step size

$\alpha \le t$ satisfies

$$f(x_k) - f(x^*) \le \frac{2\|x_0 - x^*\|^2}{\alpha(k+1)^2}$$

and the same result holds for backtracking, with α replaced by t

Acceleration can be a very effective speedup tool, but it can be disadvantageous sometimes.

In practice, the speedup of using acceleration is diminished in the presence of warm starts.

Example: Suppose we want to solve lasso problem for tuning parameters values $\lambda_1 > \lambda_2 > \cdots > \lambda_r$

- When solving for λ_1, initialize $x_0 = 0$, record solution $\hat{x}(\lambda_1)$
- When solving for λ_j, initialize $x_0 = \hat{x}(\lambda_{j-1})$, then record the solution for λ_{j-1}

Over a fine enough grid of λ values, proximal gradient descent can often perform just as well without acceleration.

Example: Recall matrix completion problem, the proximal gradient update is $B_{k+1} = S_{\lambda\alpha}(P_\Omega(Y) + P_{\Omega^\perp}(B))$, where $S_{\lambda\alpha}$ is the matrix soft-thresholding operator. This update requires SVD.

- Acceleration changes the argument we pass to the proximal mapping to $v - \alpha\nabla g(v)$ instead of $B_k - \alpha\nabla g(B_k)$. For matrix completion and $\alpha = 1$,

$$B - \nabla g(B) = \underbrace{P_\Omega(Y)}_{\text{sparse}} + \underbrace{P_{\Omega^\perp}(B)}_{\text{low rank}} \Rightarrow \text{fast SVD}$$

$$v - \nabla g(v) = \underbrace{P_\Omega(Y)}_{\text{sparse}} + \underbrace{P_{\Omega^\perp}(v)}_{\text{not necessarily low rank}} \Rightarrow \text{slow SVD}$$

- In this case, backtracking may not be a good idea either. One backtracking loop evaluates generalized gradient $G_\alpha(x)$, i.e., evaluate $\text{prox}_\alpha(x)$, across various values of α. For matrix completion, this means multiple SVD.

※ Nov. 9th 2021

Proximal Newton method

Recall motivation for proximal gradient descent: iteratively minimizing quadratic expansion in g, plus h

$$x_{k+1} = \arg\min_z \frac{1}{2\alpha}\|x_k - \alpha\nabla g(x_k) - z\|_2^2 + h(z)$$
$$= \arg\min_z \nabla g(x_k)^T(z - x_k) + \frac{1}{2\alpha}\|x_k - z\|_2^2 + h(z)$$

Quadratic approximation uses $\frac{1}{\alpha}I$ for the hessian

For proximal Newton method: We repeat

$$v_{k+1} = \arg\min_v \nabla g(x_k)^T v + \frac{1}{2}v^T H_k v + h(v + x_k)$$

$$x_{k+1} = x_k + \alpha_k v_{k+1}$$

Here $H_k = \nabla^2 g(x_k)$ is the Hessian at x_k, and α_k is a step size

Equivalent formulation

$$z_{k+1} = \arg\min_z \nabla g(x_k)^T(z - x_k) + \frac{1}{2}(z - x_k)^T H_k(z - x_k) + h(z)$$

$$x_{k+1} = x_k + \alpha_k(z_{k+1} - x_k)$$

Given a positive definite matrix H, define

$$\text{prox}_H(x) = \arg\min_z \frac{1}{2}\|x - z\|_H^2 + h(z)$$

This is called a <u>scaled proximal mapping</u>

With $H = \frac{1}{\alpha}I$, we get back the usual definition of proximal mapping

The scaled proximal mapping retains many of the nice properties of the usual proximal mapping (e.g. uniqueness,

nonexpansiveness)

Now consider

$$z_{k+1} = \operatorname{argmin}_z \nabla g(x_k)^T(z - x_k) + \frac{1}{2}(z - x_k)^T H_k (z - x_k) + h(z)$$

$$= \operatorname{argmin}_z \frac{1}{2}\|x_k - H_k^{-1}\nabla g(x_k) - z\|_{H_k}^2 + h(z)$$

Another equivalent form for proximal Newton update

$$z_{k+1} = \text{prox}_{H_k}(x_k - H_k^{-1}\nabla g(x_k))$$

$$x_{k+1} = x_k + \alpha_k(z_{k+1} - x_k)$$

Notes:
- When $h=0$, we get back the usual Newton update
- If we replace H_k by $\frac{1}{\alpha}I$, and set $\alpha_k = 1$, we get proximal gradient descent, with step size α.
- Difficulty of getting the solution to the proximal mapping depends strongly on h. However, now it also depends on the structure of the Hessian of g. For example, having diagonal or banded Hessian generally makes a big difference compared to a dense Hessian.

Step size: We apply the following backtracking line search:

fix $0 < c_1 \leq \frac{1}{2}$, $0 < \rho < 1$, and let

$$v = \text{prox}_H(x - H^{-1}\nabla g(x)) - x$$

be the proximal Newton direction at a given iteration.

Start with $\alpha = 1$, and while

$$f(x + \alpha v) > f(x) + c_1 \alpha \nabla g(x)^T v + c_1(h(x + \alpha v) - h(x))$$

we shrink $\alpha = \rho\alpha$.

Note: It avoids recomputing the proximal mapping at each backtracking iterations, which is a different scheme compared to the proximal gradient descent.

Comparison between proximal gradient descent and proximal Newton method:

proximal gradient	proximal Newton
· Iteratively minimizing $\|b-x\|_2^2+h(x)$	· Iteratively minimizing $b^Tx+\frac{1}{2}x^TAx+h(x)$
· Often closed form solution to the prox	· Almost never closed-form solution to the prox
· Iterations are cheap	· Iterations are very, very expensive
· Convergence of gradient descent	· Convergence of Newton's method

So we use proximal Newton when we have a faster optimizer for the scaled proximal mapping, and expect few iterations.

Convergence analysis (follows Lee et al. 2014)

Assume that $f=g+h$, where g,h convex and $g\in C^2$. Assume further · $mI\leq\nabla^2 g\leq LI$, and $\nabla^2 g$ is Lipschitz with M.

· $\text{prox}_H(\cdot)$ is exactly evaluated.

Theorem: Suppose $\{x_k\}$ is generated from proximal Newton method with backtracking line search, for all $k\geq k_0$,

$$\|x_k-x^*\|_2\leq\frac{M}{2m}\|x_{k-1}-x^*\|_2^2$$

After $k\geq k_0$, to get within $f(x_k)-f(x^*)\leq\varepsilon$, we need $O(\log\log\frac{1}{\varepsilon})$ iterations.

Two notable examples of proximal Newton method

· glmnet (Friedman et al. 2009) proximal Newton for ℓ_1 penalty generalized linear models. The scaled proximal mapping is evaluated using coordinate descent.

· QUIC (Hsiech et al. 2011): proximal Newton for graphical lasso

problem uses factorization tricks, the scaled proximal mapping is evaluated using coordinate descent.

Note: With proximal Newton, we essentially always perform inexact solution to the proximal mapping evaluation

For large problems, computing the Hessian is prohibitive.

Proximal quasi-Newton avoids forming $H_k = \nabla^2 g(x_k)$ at each step.

- Lee et al. (2014) proposes BFGS-type update rules. These work very well empirically, and achieve local super-linear convergence
- Tseng and Yun (2009) considered smooth plus block separable problem, propose approximating the Hessian is a blockwise fashion. Their method has linear convergence.

Quasi-Newton can be helpful not only when Hessian is burdensome computationally, but also when it is ill-conditioned, singular or near singular.

Projected Newton

When $h = I_C(x)$, C is convex, our problem becomes

$$\min_x g(x) \text{ subject to } x \in C.$$

Proximal Newton update

$$z_{k+1} = \arg\min_{z \in C} \frac{1}{2} \| x_k - H_k^{-1} \nabla g(x_k) - z \|_{H_k}^2$$

$$= \arg\min_{z \in C} \nabla g(x_k)^T (z - x) + \frac{1}{2} (z - x)^T H_k (z - x)$$

When $H = I$, this is a projection of $x_k - H_k^{-1} \nabla g(x_k)$ onto C, but not a projection in general. In fact, it is usually much more complicated. Hence while proximal gradient descent in this case reduces to projected gradient descent, projected Newton does not follow

from proximal Newton.

Projected Newton method can be made to work for box constraints.

(Bertsekas, 1982, "projected Newton method for optimization problems with simple constraints"

Schmidt, Kim and Sra, 2011, "Projected Newton-type methods in machine learning"

Kim, Sra and Dhillon, 2010, "Tackling box-constrained optimization via a new projected quasi-Newton approach")

Consider $\min_{x \in \mathbb{R}^n} g(x)$ subject to $l_i \le x_i \le u_i$, $i = 1, \cdots, n$

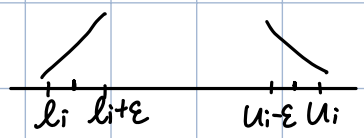
x_0: initial point

$\varepsilon > 0$ small constant

Repeats the following steps for $k = 0, 1, 2, \cdots$

- Define the binding set

$$B_k = \{ i : x_i^{(k)} \le l_i + \varepsilon \text{ and } \nabla_i g(x^{(k)}) > 0 \} \cup \{ i : x_i^{(k)} \ge u_i - \varepsilon \text{ and } \nabla_i g(x^{(k)}) < 0 \}$$



- Define the free set

$$F_k = \{1, \cdots, n\} \setminus B_k$$

- Define the inverse of the principal submatrix of the Hessian along the free variables

$$S^{(k)} = [(\nabla^2 g(x^k))_{F_k}]^{-1}$$

- Take a Newton step along the free variables only, then project

$$x^{(k+1)} = P_{[l,u]}(x^{(k)} - \alpha_k \begin{bmatrix} S^{(k)} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \nabla_{F_k} g(x^k) \\ \nabla_{B_k} g(x^k) \end{bmatrix}),$$

where $P_{[l,u]}$ is the projection onto $[l,u] = [l_1, u_1] \times \cdots \times [l_n, u_n]$.

- Note that the update leaves binding set effectively untouched

Convergence: Bertsekas (1982) shows that, under appropriate assumptions

projected Newton achieves super-linear convergence rate

Applications: Nonnegative least squares, support vector machine dual, graphical lasso dual,

※ Nov. 16th 2021

Theory of constrained optimization (Chap. 12 in the textbook)

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{aligned} c_i(x)&=0 \quad i\in\mathcal{E} \\ c_i(x)&\geq 0 \quad i\in\mathcal{I}\end{aligned}$$

$\Omega=\{x: x \text{ satisfies all the constraints}\}$ → feasible set

Def. The active set $\mathcal{A}(x)$ at any feasible x consists of the equality constraint indices from $\mathcal{E}$ together with the indices of the inequality constraints i for which $c_i(x)=0$, i.e.,

$$\mathcal{A}(x)=\mathcal{E}\cup\{i\in\mathcal{I} \mid c_i(x)=0\}$$

Focus our attention on constrained optimization with inequality constraints: $\min_{x\in\mathbb{R}^n} f(x)$ subject to $c_i(x)\geq 0$, $i\in\mathcal{I}$.

Def. A vector $d\in\mathbb{R}^n, d\neq 0$ is said to be a feasible direction at x if there exists $\delta_1>0$ such that $x+\alpha d$ satisfies all constraints for all $\alpha\in(0,\delta_1)$

Let $F(x)$ = Set of feasible directions at $x\in\Omega$.

Def. A vector $d\in\mathbb{R}^n, d\neq 0$ is said to be a descent direction at x if there exists $\delta_2>0$ such that $f(x+\alpha d)<f(x)$ for all $\alpha\in(0,\delta_2)$

Let $D(x)$ = Set of descent directions at $x\in\Omega$.

Theorem. x^* is a local minimum of f, then $F(x^*)\cap D(x^*)=\emptyset$. (*)

Note that $\nabla f^T(x)d<0 \Rightarrow f(x+\alpha d)<f(x)$ for some $\alpha\in(0,\delta_2)$

$\Rightarrow d$ is a descent direction $\Rightarrow d\in D(x)$

$\tilde{D}(x) = \{d: \nabla f(x)^T d < 0\} \subset D(x)$ ($\stackrel{\text{def}}{=}$ any direction if $A(x)=\phi$)

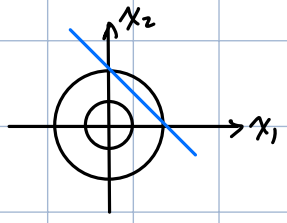
Lemma: For any $x \in \Omega$, $\tilde{F}(x) \stackrel{\text{def}}{=} \{d: \nabla g_j(x)^T d > 0, j \in I \cap A(x)\} \subset F(x)$

Then $x^* \in \Omega$ is a local minimum if $\tilde{F}(x^*) \cap \tilde{D}(x^*) = \phi$

- This is only a necessary condition for a local minimum.
- Cannot be directly used for equality constrained problem.

e.g. $\min_{\binom{x_1}{x_2} \in \mathbb{R}^2} x_1^2 + x_2^2$, s.t. $x_1 + x_2 = 1 \iff \begin{cases} x_1 + x_2 \geq 1 \\ -x_1 - x_2 \geq -1 \end{cases}$

$c_1(x) = x_1 + x_2$, $c_2(x) = -x_1 - x_2$



$\tilde{F}(x) = \{d: \nabla c_1(x)^T d > 0, \nabla c_2(x)^T d > 0\}$

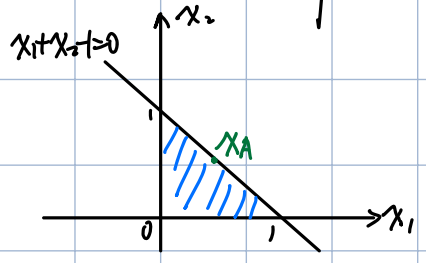
$= \{d: \binom{1}{1}^T d > 0, \binom{-1}{-1}^T d > 0\} = \phi$

- Utility of this condition depends on the constraint representation

e.g. $\min_{\binom{x_1}{x_2} \in \mathbb{R}^2} x_1^2 + x_2^2$ s.t. $-(x_1 + x_2 - 1)^3 \geq 0$

$x_1 \geq 0$

$x_2 \geq 0$



$\tilde{F}(x_A) = \{d: \nabla c_1(x_A)^T d > 0\} = \phi$

$\nabla c_1(x_A) = \begin{pmatrix} -3(x_1+x_2-1)^2 \\ -3(x_1+x_2-1)^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Represent the constraint

$\min_{\binom{x_1}{x_2} \in \mathbb{R}^2} x_1^2 + x_2^2$ s.t. $-(x_1 + x_2 - 1) \geq 0$

$x_1 \geq 0$

$x_2 \geq 0$

$\tilde{F}(x_A) = \{d: \nabla c_1(x_A)^T d > 0\} = \{d: \binom{-1}{-1}^T d > 0\}$

($\tilde{F}(x^*) \cap \tilde{D}(x^*) = \phi$)

Write the condition (*) in a compact way

Let $A = \begin{pmatrix} \nabla f(x^*)^T \\ \hline -\nabla g_j(x^*)^T, j \in A(x^*) \end{pmatrix}_{(1+|A(x^*)|) \times n}$ (cardinality of $A(x^*)$)

$x^* \in X$ is a local minimum $\Rightarrow \{d: Ad < 0\} = \phi$.

Farka's Lemma: Let $A \in \mathbb{R}^{m \times n}$ and $c \in \mathbb{R}^n$. Then, exactly one of the following two systems has a solution

(I) $Ax \leq 0$, $c^T x > 0$ for some $x \in \mathbb{R}^n$

(II) $A^T y = c$, $y \geq 0$ for some $y \in \mathbb{R}^m$

Corollary: Let $A \in \mathbb{R}^{m \times n}$. Then exactly one of the following two systems has a solution

(I) $Ax < 0$ for some $x \in \mathbb{R}^n$

(II) $A^T y = 0$, $y \geq 0$ for some nonzero $y \in \mathbb{R}^m$

$x^* \in \Omega$ is a local minimum $\Rightarrow \{d : Ad < 0\} = \phi$

$\Rightarrow \exists \lambda_0 \geq 0$, and $\lambda_j \geq 0$, $j \in A(x^*)$ (not all λ's are 0), such that

$$\lambda_0 \nabla f(x^*) - \sum_{j \in A(x^*)} \lambda_j \nabla c_j(x^*) = 0$$

Assume $\nabla c_j(x)$ are linear independent

Def: A point x^* is said to be a regular point if the gradient vectors $\nabla c_j(x^*)$, $j \in A(x^*)$ are linearly independent.

x^* is a regular point $\Rightarrow \lambda_0 \neq 0$

Let $\lambda_j^* = \lambda_j / \lambda_0$, $j \in A(x^*)$, set $\lambda_j^* = 0$ if $j \notin A(x^*)$

$$\nabla f(x^*) - \sum_{j \in I} \lambda_j^* \nabla c_j(x^*) = 0$$

$$\lambda_j^* c_j(x^*) = 0, \quad \forall j \in I$$

$$\lambda_j^* \geq 0, \quad \forall j \in I$$

Karush-Kuhn-Tucker

KKT necessary condition: If x^* is a local minimum and a regular point, then there exists a unique vector $\lambda^* = (\lambda_1^*, \cdots, \lambda_\ell^*)^T$, $\ell = |I|$ such that $\nabla f(x^*) - \sum_{j \in I} \lambda_j^* \nabla c_j(x^*) = 0$

$$\lambda_j^* c_j(x^*) = 0, \quad \forall j \in I$$

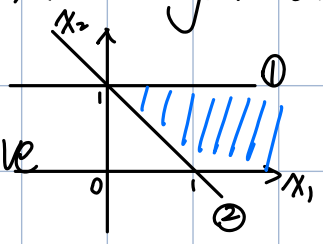
complementary slackness condition

$$\lambda_j^* \geq 0, \quad \forall j \in I$$

KKT point: (x^*, λ^*), $x^* \in \Omega$, $\lambda^* \geq 0$

- At a local minimum, active set is unknown, investigate all active sets

Ex: $\min_{\binom{x_1}{x_2}\in\mathbb{R}^2} x_1^2+x_2^2$ s.t. $x_2\le 1$, $x_1+x_2\ge 1$



· Case 1: Constraints 1 and 2 are both active

$x^*=\binom{0}{1}$

$L=x_1^2+x_2^2-\lambda_1(1-x_1)-\lambda_2(x_1+x_2-1)$

$\nabla_x L(x^*,\lambda^*)=0$

$\nabla_x L(x,\lambda)=\binom{2x_1}{2x_2}-\lambda_1\binom{0}{-1}-\lambda_2\binom{1}{1}$

$\nabla_x L(x^*,\lambda^*)=\binom{0}{2}-\lambda_1^*\binom{0}{-1}-\lambda_2^*\binom{1}{1}\overset{\text{set}}{=}0 \Rightarrow \begin{cases}\lambda_1^*=-2\\ \lambda_2^*=0\end{cases}$, (x^*,λ^*) is not a KKT point

· Case 2: Constraint 2 is active

Constraint 1 is inactive $\Rightarrow \lambda_1^*=0$

$\nabla f(x^*)-\lambda_2^*\nabla C_2(x^*)=\binom{0}{0}$

$\Leftrightarrow \binom{2x_1^*}{2x_2^*}-\lambda_2^*\binom{1}{1}=\binom{0}{0} \Rightarrow x_1^*=x_2^*=\frac{1}{2}\lambda_2^*$

$x_1^*+x_2^*-1=0$

$\Rightarrow \lambda_2^*=1,\ x_1^*=\frac{1}{2},\ x_2^*=\frac{1}{2}$ KKT point

· Case 3: Constraint 1 is active

Constraint 2 is inactive ...

· Case 4: Both constraints are inactive ...

· A KKT point can be a local maximum

Ex: $\min -x^2$ s.t. $x\le 0$

$L(x,\lambda)=-x^2-\lambda(-x)$

$\frac{\partial L}{\partial x}=-2x+\lambda=0$

· Case 1: At $x^*=0$, the constraint is active

Then $\lambda^*=0$

(x^*,λ^*) is a KKT point (local maximum)

· Case 2: At $x^*\neq 0$, the constraint is inactive

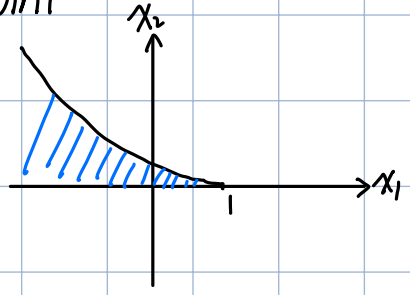
Then $\lambda^*=0$

We have $x^* = 0$

There is no KKT point in this case.

- Every local minima need not be a KKT point

Ex: $\min_{(x_1,x_2)\in\mathbb{R}^2} -x_1$ s.t. $(1-x_1)^3 - x_2 \geq 0$, $x_2 \geq 0$



$x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$L(x,\lambda) = -x_1 - \lambda_1((1-x_1)^3 - x_2) - \lambda_2 x_2$

$\nabla_x L(x,\lambda) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \lambda_1 \begin{pmatrix} -3(1-x_1)^2 \\ -1 \end{pmatrix} - \lambda_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$\nabla_x L(x^*,\lambda^*) = \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \lambda_1^* \begin{pmatrix} 0 \\ -1 \end{pmatrix} - \lambda_2^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ \lambda_1^* + \lambda_2^* \end{pmatrix} \neq 0$

$\nabla c_1(x^*)$ $\nabla c_2(x^*)$ → not linearly independent

→ x^* is not a regular point

<u>Linear independent constraint qualification (LICQ)</u>

$\nabla c_j(x^*)$, $j \in \mathcal{A}(x^*)$ are linearly independent

<u>Mangasarian-Fromovitz constraint qualification (MFCQ)</u>

$\{d : \nabla c_j(x^*)^T d > 0,\ j \in \mathcal{A}(x^*)\} \neq \emptyset$

- MFCQ is weaker than LICQ

If LICQ is satisfied, then MFCQ is satisfied

△Constraint qualification ensures that the linearized approximation is reasonable (captures the shape of Ω in a neighborhood of x^*)

※ Nov. 18th 2021

<u>Convex problem</u>

Consider the problem $\min_{x\in\mathbb{R}^n} f(x)$, $c_i \geq 0$, $i \in I$

If $f, -c_i, i \in I$ are differentiable convex function, then this is a convex program.

- $\Omega = \{x \in \mathbb{R}^n : c_i(x) \geq 0,\ i \in I\}$ is a convex set

- Every local minimum of a convex problem is a glocal minimum
- The set of all optimal solutions to a convex problem is convex

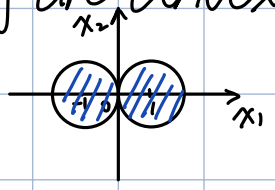
Theorem: If $x^* \in \Omega$ is a regular point, then for x^* to be a global minimum of convex program, first order KKT conditions are necessary and sufficient

Slater's constraint qualification: There exists $y \in \Omega$ such that

$$c_i(y) > 0, \text{ for all } i \in I.$$

Useful when the constraint functions $-c_j$ are convex.

Ex: $\min_{(x_1,x_2)\in\mathbb{R}^2} x_1 + x_2$ s.t. $-(x_1-1)^2 - x_2^2 \geq -1$, $-(x_1+1)^2 - x_2^2 \geq -1$



This is a convex program

$x^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ is the global minimum

$$L(x,\lambda) = x_1 + x_2 + \lambda_1((x_1-1)^2 + x_2^2 - 1) + \lambda_2((x_1+1)^2 + x_2^2 - 1)$$

$$\frac{\partial L}{\partial x}(x,\lambda) = \begin{pmatrix} 1 + 2\lambda_1(x_1-1) + 2\lambda_2(x_1+1) \\ 1 + 2\lambda_1 x_2 + 2\lambda_2 x_2 \end{pmatrix}$$

$$\frac{\partial L}{\partial x}(x,\lambda)\Big|_{x=x^*} = \begin{pmatrix} 1 + 2\lambda_2 - 2\lambda_1 \\ 1 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

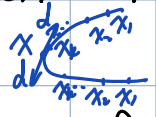
x^* is not a KKT point

It does not satisfy the Slater's constraint qualification.

Consider the problem

$$\min_{x\in\mathbb{R}^n} f(x), \quad c_i(x) = 0, \ i \in \mathcal{E}$$

Def: A vector $d \in \mathbb{R}^n$ is said to be a tangent of Ω at x if either $d = 0$ or there exists a sequence $\{x_k\} \subset \Omega$, $x_k \neq x$ such that $x_k \to x$, $\frac{x_k - x}{\|x_k - x\|} \to \frac{d}{\|d\|}$



The collection of all tangents of Ω at x is called the tangent set at x and is denoted by $T(x)$

Def. A point $x \in \Omega$ is said to be a regular point if $\nabla c_i(x), i \in \mathcal{E}$ are linearly independent.

Theorem. If x^* is a regular point with respect to the constraints $c_i(x), i \in \mathcal{E}$ and x^* is a local minimum (or maximum) of f subject to these constraints, then $\nabla f(x^*)$ is orthogonal to the tangent set $T(x^*)$.

Let $x(t)$ be any smooth curve s.t. $x(t) \in \Omega$, $x(0) = x^*$, $\left.\frac{dx(t)}{dt}\right|_{t=0} = d$, x^* is a local minimum (or maximum)

$\left.\frac{d}{dt} f(x(t))\right|_{t=0} = 0 \Rightarrow \nabla f(x^*)^T d = 0$, $d \in T(x)$

Theorem. Let x^* be a regular point and a local minimum. Then there exists $\mu_i^* \in \mathbb{R}$ such that

$$\nabla f(x^*) - \sum_{i \in \mathcal{E}} \mu_i^* \nabla c_i(x^*) = 0$$

Note that we do not require $\mu_i^* \geq 0$ as in the inequality constraints.

General constrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \begin{array}{l} c_i(x) \geq 0, \ i \in \mathcal{I} \\ c_i(x) = 0, \ i \in \mathcal{E} \end{array}$$

x^* is a regular point if $\{\nabla c_i(x), i \in \mathcal{A}(x) \cap \mathcal{I}\} \cup \{\nabla c_i(x), i \in \mathcal{E}\}$ is a set of linearly independent vectors.

KKT necessary conditions: If x^* is a local minimum and a regular point, then there exists unique scalars $\lambda_i^* \geq 0$ and $\mu_i^* \in \mathbb{R}$ s.t.

$$\begin{cases} \nabla f(x^*) - \sum_{i \in \mathcal{I}} \lambda_i^* \nabla c_i(x^*) - \sum_{i \in \mathcal{E}} \mu_i^* \nabla c_i(x^*) = 0 \\ \lambda_i^* \nabla c_i(x^*) = 0, \ i \in \mathcal{I} \\ \lambda_i^* \geq 0, \ i \in \mathcal{I} \end{cases}$$

- KKT point (x^*, λ^*, μ^*) satisfying above conditions

· First order KKT conditions also satisfied at a local maxima.

For convex program

Assumptions: f and $-c_i$'s in inequality constraints are convex

· c_i's in the equality constraints satisfy $c_i(x) = a_i^T x - b_i$

· Slater's constraint qualification holds for Ω.

Then the first order KKT conditions are necessary and sufficient for a global minimum of a convex program.

Pf: Since f and $-c_i$, $i \in I$ are convex functions

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

$$-c_i(x) \geq -c_i(x^*) - \nabla c_i(x^*)^T (x - x^*)$$

For $i \in \mathcal{E}$, $c_i(x) = c_i(x^*) + \nabla c_i(x^*)^T (x - x^*)$

$$f(x) \geq f(x) - \sum_{i \in I} \lambda_i^* c_i(x) - \sum_{i \in \mathcal{E}} \mu_i^* c_i(x)$$

$$\geq f(x^*) - \sum_{i \in I} \lambda_i^* c_i(x^*) - \sum_{i \in \mathcal{E}} \mu_i^* c_i(x^*)$$

$$+ \underbrace{\left(\nabla f(x^*) - \sum_{i \in I} \lambda_i^* \nabla c_i(x^*) - \sum_{i \in \mathcal{E}} \mu_i^* \nabla c_i(x^*)\right)^T (x - x^*)}_{\nabla_x L(x^*, \lambda^*, \mu^*) = 0}$$

So we have $f(x) \geq f(x^*)$

Nov. 23th 2021

Consider the problem

$$\min_x f(x) \quad \text{s.t.} \quad c_i(x) \geq 0, \ i \in I$$

$$c_i(x) = 0, \ i \in \mathcal{E}$$

Assume that x^* is a regular point (LICQ)

($\{\nabla c_i(x^*), i \in I \cap \mathcal{A}(x^*)\} \cup \{\nabla c_i(x^*), i \in \mathcal{E}\}$ is a set of linearly independent vectors)

$$L(x, \lambda, \mu) = f(x) - \sum_{i \in I} \lambda_i c_i(x) - \sum_{i \in \mathcal{E}} \mu_i c_i(x)$$

KKT necessary conditions (second order)

If x^* is a local minimum and a regular point, then there exists scalars $\lambda_i^* \geq 0$ and $\mu_i^* \in \mathbb{R}$ such that

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0$$

$$\lambda_i^* c_i(x^*) = 0 \quad , \forall i \in I$$

$$\lambda_i^* \geq 0 \quad , \forall i \in I$$

and $d^T \nabla_{xx}^2 L(x^*, \lambda^*, \mu^*) d \geq 0$ for all $d \in C(x^*, \lambda^*, \mu^*)$

$C(x^*, \lambda^*, \mu^*)$ is called the <u>critical cone</u>

$$d \in C(x^*, \lambda^*, \mu^*) \Leftrightarrow \begin{cases} \nabla c_i(x^*)^T d = 0 \quad , \forall i \in \mathcal{E} \\ \nabla c_i(x^*)^T d = 0 \quad , \forall i \in \mathcal{A}(x^*) \cap I \text{ with } \lambda_i^* > 0 \\ \nabla c_i(x^*)^T d \geq 0 \quad , \forall i \in \mathcal{A}(x^*) \cap I \text{ with } \lambda_i^* = 0 \end{cases}$$

The critical cone contains directions that would tend to "adhere" to the active inequality constraints even when we were to make small changes to the objective, as well as to the equality constraints.

KKT sufficient conditions (second order)

If there exists x^*, $\lambda_i^* \geq 0$, and $\mu_i^* \in \mathbb{R}$ such that

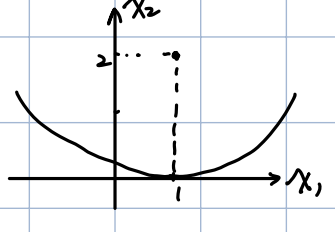
$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0$$

$$\lambda_i^* c_i(x^*) = 0 \quad , \forall i \in I$$

$$\lambda_i^* \geq 0 \quad , \forall i \in I$$

and $d^T \nabla_{xx}^2 L(x^*, \lambda^*, \mu^*) d > 0$ for all $d \neq 0$ and $d \in C(x^*, \lambda^*, \mu^*)$ then x^* is a strict local minimum.

Ex. Find the point on the parabola $x_2 = \frac{1}{5}(x_1 - 1)^2$ that is closest to $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ in the Euclidean norm sense.



$$\min_x (x_1 - 1)^2 + (x_2 - 2)^2 \text{ subject to } (x_1 - 1)^2 - 5x_2 = 0$$

$$L(x, \mu) = (x_1 - 1)^2 + (x_2 - 2)^2 - \mu[(x_1 - 1)^2 - 5x_2]$$

$$\nabla_x L(x, \mu) = \begin{pmatrix} 2(x_1 - 1) - 2\mu(x_1 - 1) \\ \end{pmatrix} = \begin{pmatrix} (2 - 2\mu)(x_1 - 1) \\ \end{pmatrix} \underline{\text{set}} \; 0$$

$\left(2(x_2-2)+5\mu \right)$ $\left(2(x_2-2)+5\mu \right)$

Case 1: $x_1^*=1 \Rightarrow x_2^*=0 \Rightarrow \mu^*=\frac{4}{5}$

$\nabla^2_{xx}L(x,\mu)=\begin{pmatrix}2-2\mu & 0\\ 0 & 2\end{pmatrix}$

$\nabla^2_{xx}L(x^*,\mu^*)=\begin{pmatrix}\frac{2}{5} & 0\\ 0 & 2\end{pmatrix}$

$C_1(x)=(x_1-1)^2-5x_2 \quad \nabla C_1(x)=\begin{pmatrix}2(x_1-1)\\ -5\end{pmatrix} \quad \nabla C_1(x^*)=\begin{pmatrix}0\\ -5\end{pmatrix}$

$C(x^*,\mu^*)=\{d\neq 0 : \nabla C_i(x^*)^T d=0\}$

$=\{d\neq 0 : \begin{pmatrix}0\\ -5\end{pmatrix}^T d=0\}$

$=\{\begin{pmatrix}d_1\\ 0\end{pmatrix} : d\in\mathbb{R}, d_1=0\}$

$\begin{pmatrix}d_1\\ 0\end{pmatrix}^T\begin{pmatrix}\frac{2}{5} & 0\\ 0 & 2\end{pmatrix}\begin{pmatrix}d_1\\ 0\end{pmatrix}=\frac{2}{5}d_1^2>0$

Case 2: $\mu^*=1 \quad x_2^*=-0.5$

This is not a feasible point

Caution: ① $\min_x (x_1-1)^2+(x_2-2)^2$ s.t. $(x_1-1)^2-5x_2=0$

② $\min_{x_2} 5x_2+(x_2-2)^2$

Are ① and ② equivalent? No

① is equivalent to $\min_{x_2} 5x_2+(x_2-2)^2$ s.t. $x_2\geq 0$

But you can convert to $\min_{x_1} (x_1-1)^2+\left(\frac{(x_1-1)^2}{5}-2\right)^2$

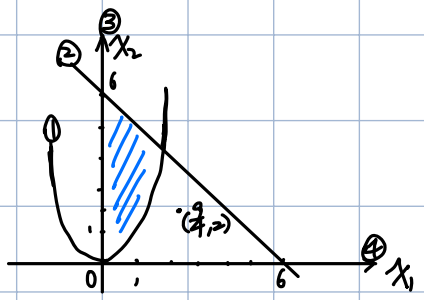
Let $y=x_1-1$, $\min_y y^2+\left(\frac{y^2}{5}-2\right)^2$

Ex. $\min_x (x_1-\frac{9}{4})^2+(x_2-2)^2$ s.t. $x_2-x_1^2\geq 0$

$6-x_1-x_2\geq 0$

$x_1\geq 0$

$x_2\geq 0$



convex program

Slater's constraint qualification is satisfied

$L(x,\lambda)=(x_1-\frac{9}{4})^2+(x_2-2)^2-\lambda_1(x_2-x_1^2)-\lambda_2(6-x_1-x_2)-\lambda_3x_1-\lambda_4x_2$

Case 1: ① is active, ②③④ are inactive

$\lambda_2 = \lambda_3 = \lambda_4 = 0$

$$\nabla_x L(x,\lambda) = \begin{pmatrix} 2(x_1 - \frac{9}{4}) + 2\lambda_1 x_1 \\ 2(x_2 - 2) - \lambda_1 \end{pmatrix} \overset{\text{set}}{=} 0 \quad \left.\begin{matrix} \\ x_2 - x_1^* = 0 \end{matrix}\right\} \Rightarrow \lambda_1 = 2(x_2 - 2) = 2(x_1^2 - 2)$$

$$x_1 - \frac{9}{4} + 2x_1^3 - 4x_1 = 0$$

$$2x_1^3 - 3x_1 - \frac{9}{4} = 0$$

$$2(x_1 - \tfrac{3}{2})(x_1^2 + \tfrac{3}{2}x_1 + \tfrac{3}{4}) = 0$$

$$x^* = \begin{pmatrix} \frac{3}{2} \\ \frac{9}{4} \end{pmatrix} \quad \lambda^* = \frac{1}{2}$$

$$\nabla_{xx} L(x,\lambda) = \begin{pmatrix} 2 + 2\lambda_1 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\nabla C_1(x) = \begin{pmatrix} -2x_1 \\ 1 \end{pmatrix} \quad \nabla C_1(x^*) = \begin{pmatrix} -3 \\ 1 \end{pmatrix}$$

$$C(x^*,\lambda^*) = \{d \neq 0 : \nabla C_1(x^*)^T d = 0\} = \{d \neq 0 : -3d_1 + d_2 = 0\}$$

$$= \left\{ \begin{pmatrix} d_1 \\ 3d_1 \end{pmatrix} : d_1 \in \mathbb{R}, d \neq 0 \right\}$$

$$d^T \nabla_{xx} L(x,\lambda) d = 3d_1^2 + 18d_1^2 = 21d_1^2 > 0$$

$x^* = \begin{pmatrix} 3/2 \\ 9/4 \end{pmatrix}$ is a local minimum

Case2: ① and ② are active, ③ and ④ are inactive

$\lambda_3 = \lambda_4 = 0$

$$x^* = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$$

$$\left.\begin{matrix} 2(2 - \frac{9}{4}) + 2\lambda_1 \cdot 2 + \lambda_2 = 0 \\ 2(4-2) - \lambda_1 + \lambda_2 = 0 \end{matrix}\right\} \Rightarrow \begin{cases} -\frac{1}{2} + 4\lambda_1 + \lambda_2 = 0 \\ 4 - \lambda_1 + \lambda_2 = 0 \end{cases} \Rightarrow \begin{cases} \lambda_1^* = \frac{9}{10} \\ \lambda_2^* = \frac{-31}{10} < 0 \end{cases}$$

$x^* = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ is not a KKT point

Case3: ③ is active, ①②④ are inactive

$\lambda_1 = \lambda_2 = \lambda_4 = 0$

$$\left.\begin{matrix} 2(x_1 - \frac{9}{4}) - \lambda_3 = 0 \\ 2(x_2 - 2) = 0 \\ x_1 = 0 \end{matrix}\right\} \Rightarrow \begin{cases} x_1^* = 0 \\ x_2^* = 2 \\ \lambda_3^* = -\frac{9}{2} \end{cases}$$

$x^* = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ is not a KKT point

Case4: ② active, ①③④ inactive

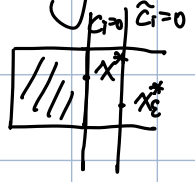
$\lambda_1 = \lambda_3 = \lambda_4 = 0$

$x^* = \begin{pmatrix} 25/8 \\ 23/8 \end{pmatrix}$ $\lambda_2^* = -\frac{7}{4}$

x^* is not a KKT point

Note: If x^* is not a regular point, the Lagrange Multipliers may not exist, or multiple Lagrange multipliers

- $f(x_\varepsilon^*) - f(x^*) \approx -\lambda_i^* \varepsilon \|\nabla c_i(x^*)\|$



Duality

Primal problem $\min_{x \in X} \max_{y \in Y} \psi(x,y)$ (primal function)

Dual problem $\max_{y \in Y} \min_{x \in X} \psi(x,y)$ (dual function)

Then two problems are dual to each other

x: primal variable

y: dual variable

For any $x \in X$, $y \in Y$,

$$\min_{x \in X} \psi(x,y) \le \psi(x,y) \le \max_{y \in Y} \psi(x,y)$$

$$\max_{y \in Y} \min_{x \in X} \psi(x,y) \le \min_{x \in X} \max_{y \in Y} \psi(x,y)$$

e.g. $X = \{1,2\}$, $Y = \{1,2\}$, $\psi(x,y) = a_{x,y}$, where $A = \begin{pmatrix} -2 & 1 \\ 2 & -3 \end{pmatrix}$

Primal: $\min_x \max_y \psi(x,y) = \min_x (1,2) = 1$

Dual: $\max_y \min_x \psi(x,y) = \max_y (-2,-3) = -2$

$A = \begin{pmatrix} -2 & 1 \\ 2 & 3 \end{pmatrix}$

Primal: $\min_x \max_y \psi(x,y) = \min_x (1,2) = 1$

Dual: $\max_y \min_x \psi(x,y) = \max_y (-2,1) = 1$

△Week duality: $\max_y \min_x \psi(x,y) \le \min_x \max_y \psi(x,y)$

△Strong duality: $\max_y \min_x \psi(x,y) = \min_x \max_y \psi(x,y)$

Consider the problem

$\min_{x\in\mathbb{R}^n} f(x)$ $\quad c_i(x)\geq 0$, $i=1,2,\cdots,m$

f and $-c_i$ are all convex functions

Let $c(x)=\begin{pmatrix} c_1(x) \\ \vdots \\ c_m(x)\end{pmatrix}$

Then $L(x,\lambda)=f(x)-\lambda^T c(x)$ is the Lagrange function, where $\lambda\in\mathbb{R}^m$

x: Primal variable

λ: Dual variable

Primal function $\max_{\lambda\geq 0} L(x,\lambda)=\max_{\lambda\geq 0}(f(x)-\lambda^T c(x))$

$$=\begin{cases} f(x), & \text{if } c(x)\geq 0 \\ +\infty, & \text{otherwise}\end{cases}$$

Primal problem: $\min_{x\in\mathbb{R}^n}\max_{\lambda\geq 0} L(x,\lambda) \iff \min_{x\in\mathbb{R}^n} f(x)$ subject to $c(x)\geq 0$

Dual problem: $\max_{\lambda\geq 0}\min_{x\in\mathbb{R}^n} L(x,\lambda) \overset{\text{def}}{=} \max_{\lambda\geq 0} q(\lambda)$

For $\lambda\geq 0$, $q(\lambda)$ is the dual function: $q(\lambda)=\min_{x\in\mathbb{R}^n} L(x,\lambda)=\min_{x\in\mathbb{R}^n}(f(x)-\lambda^T c(x))$

Ex. $\min x_1^2+x_2^2+\cdots+x_n^2$ s.t. $x_1+x_2+\cdots+x_n\geq 1$

$L(x,\lambda)=x_1^2+\cdots+x_n^2-\lambda(x_1+\cdots+x_n-1)$

Dual function $q(\lambda)=\min_x L(x,\lambda)$

$x_i=\frac{\lambda}{2}$, $i=1,\cdots,n$

$q(\lambda)=\frac{n\lambda^2}{4}-\lambda\cdot n\frac{\lambda}{2}+\lambda=-\frac{n\lambda^2}{4}+\lambda$

Dual problem: $\max_{\lambda\geq 0} -\frac{n\lambda^2}{4}+\lambda$

Properties:

- The dual function is concave

$q(t\lambda_1+(1-t)\lambda_2)=\min_x L(x,t\lambda_1+(1-t)\lambda_2)$

$=\min_x\{f(x)-[t\lambda_1+(1-t)\lambda_2]^T c(x)\}$

$=\min_x[t(f(x)-\lambda_1^T c(x))+(1-t)(f(x)-\lambda_2^T c(x))]$

$$\geq \min_x t(f(x)-\lambda_1^T c(x)) + \min_x (1-t)(f(x)-\lambda_2^T c(x))$$

$$= tq(\lambda_1) + (1-t)q(\lambda_2)$$

- For any feasible $\bar{x}$ and $\bar{\lambda}\geq 0$, $q(\bar{\lambda}) \leq f(\bar{x})$ (Weak duality)
- (Strong duality)

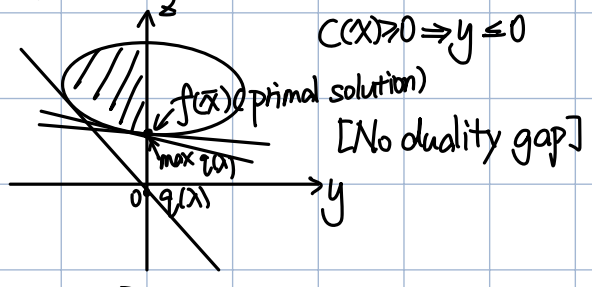
Assumption: · f and $-c_i$ are convex and continuously differentiable
· Slater's condition holds
· Let $\hat{\lambda}$ be a solution to the dual problem, and $\min_x L(x,\hat{\lambda})$ is attained at $\hat{x}$, $L(\cdot,\hat{\lambda})$ is strictly convex
· $\bar{x}$ is the solution of the primal problem

Then $\bar{x}=\hat{x}$, and $f(\bar{x})=L(\hat{x},\hat{\lambda})$

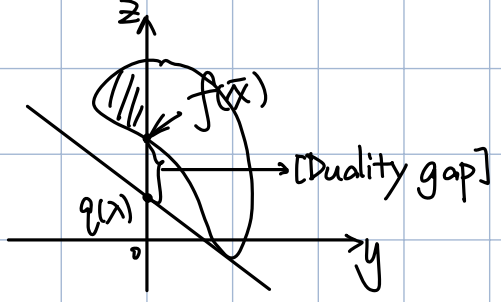
Back to the example before, we have $x^*=\frac{2}{n}$, $x_i^*=\frac{1}{n}$

Graphical interpretation.

Let $y=-c(x)$, $z=f(x)$



Dual
$\max_{\lambda\geq 0} q(\lambda)$
$q(\lambda)=\min_x f(x)-\lambda^T c(x)$
$=\min_x z+\lambda^T y$



Wolfe dual

$$\max_{x,\lambda} L(x,\lambda) \text{ subject to } \nabla_x L(x,\lambda)=0,\ \lambda\geq 0$$

Theorem. · f and $-c_i$, $i=1,2,\cdots,m$ are convex and C^1
· LICQ holds at $(\bar{x},\bar{\lambda})$, which is the solution of the primal problem

Then $(\bar{x}, \bar{\lambda})$ solves the dual problem.

Ex. Consider the linear programming problem

$\min_x c^T x$ s.t. $Ax \geq b$

$L(x, \lambda) = c^T x - \lambda^T (Ax - b)$

$\nabla_x L(x, \lambda) = c - \lambda^T A \overset{\text{set}}{=} 0 \Rightarrow A^T \lambda = c$

Wolfe dual $\max_\lambda b^T \lambda$ s.t. $A^T \lambda = c, \lambda \geq 0$

※Nov. 30th 2021

※Dec. 2nd 2021