7

UNCONSTRAINED NONLINEAR OPTIMIZATION

7.1 Optimality conditions

common notations

- x_k: kth iteration point of x.
- g_k: gradient $\nabla f(x_k)$ at x_k.
- H_k: Hessian $\nabla^2 f(x_k)$ at x_k.

7.1.1 Optimality concepts

In **unconstrained nonlinear optimization**, the goal is to generally minimize an objective function $f(x) : \mathbb{R}^N \to \mathbb{R}$ over $\mathbb{R}^N$. $f(x)$ is required to be in C_1(sometimes C_2 is required).

Example 7.1.1. Examples of objective functions include

- Linear functions such as $f(x) = cx, c \in \mathbb{R}$.
- Quadratic functions such as $f(x_1, x_2) = x_1^2 + 2x_1x_2 + 4x_2^2$.
- Polynomial functions such as $f(x) = a_0 + a_1x + a_2x_2 + \cdots a_px_p$.
- Complex nonlinear function such as the Humpback function [Figure 7.1.2] given by

$$f(x_1, x_2) = x_1^2(4 - 2.1x_1^2 + 0.33x^4) + x_1x_2 + x_2^2(-4 + 4x_2).$$

In searching for solutions to the optimization problems, we usually distinguish local and global minimums.

Definition 7.1.1 (local and global minimum). *[1, p. 5] A vector x^* is an* ***unconstrained local minimum*** *of a function $f : \mathbb{R}^n \to \mathbb{R}$ if it is no worse than its neighbors; that is, if there exist an $\delta > 0$ such that*

$$f(x) \geq f(x^*), \forall \|x - x^*\| < \delta$$

A vector x^* is an ***unconstrained global minimum*** *of f if $f(x^*) \geq f(x), \forall x \in \mathbb{R}^n$.*

Remark 7.1.1 (possible non-existence of minimizer). For some $f(x)$, the minimizer might not exist. For example, linear uncontrained optimization like $f(x) = x$ does not have a minimum. **In nature, this is because $\mathbb{R}^N$ is not a compact set.**

Local and global minimums can be further categorized into more specific types [Figure 7.1.1]:

Definition 7.1.2 (types of local minimizer). *Let a function $f(x)$ be defined in a region $D \subset \mathbb{R}^n$. We say that $x^* \in D$ is a*

- ***a global minimizer*** *of $f(x)$ in D if $f(x^*) \leq f(x)$ for all $x \in D$.*
- ***a strict global minimizer*** *of $f(x)$ in D if $f(x^*) < f(x)$ for all $x \in D, x \neq x^*$.*
- ***a local minimizer*** *of $f(x)$ in D if $f(x^*) \leq f(x)$ for all x in some open ball $B(x^*)$.*
- ***a strict local minimizer*** *of $f(x)$ in D if $f(x^*) < f(x)$ for all x in some open ball $B(x^*), x \neq x^*$.*

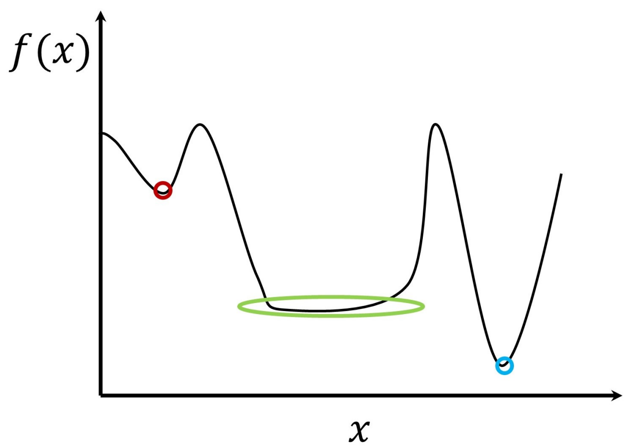


Figure 7.1.1: Demonstration of local minimizer (red, green, and blue), strict lcoal minimizer (red and blue), and global minimizer (blue).

In practice, the objective function can be a highly complex nonlinear function [Figure 7.1.2]. Searching for global minimum can be intractable. In the following, we will develop theories and algorithms can allow us to find one local minimum.

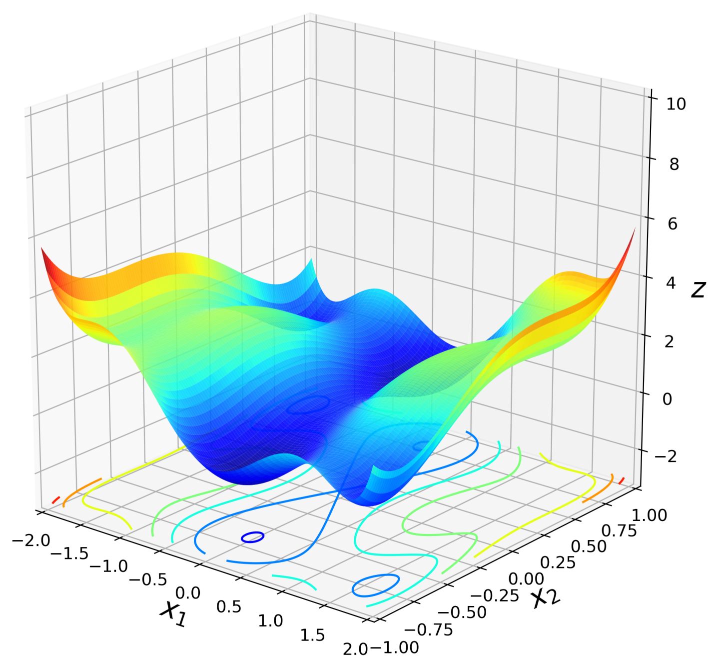


Figure 7.1.2: A complex objective function in unconstrained optimization.

7.1.2 Necessary and sufficient conditions

Theorem 7.1.1 (first order necessary condition). *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, if x^* is local minimizer, then $\nabla f = 0$.*

Proof. Use contradiction to prove, if $g = \nabla f \neq 0$, and assume $\nabla f > 0$, then we can show

$$f(x^* - ag(x^*)) = f(x^*) - a\|g(x^*)\|^2 + O(a^2).$$

Then for sufficiently small a, we have

$$f(x^* - ag(x^*)) - f(x^*) < 0,$$

which contradicts with the fact that $f(x^*)$ is local minimum. □

Remark 7.1.2. This is not a sufficient condition. Consider a counter example is $f(x) = x^3$. At $x = 0$, $f'(0) = 0$ but it is not a local minimum.

Theorem 7.1.2 (second order necessary condition). *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, if x^* is local minimizer, then $\nabla^2 f(x^*) = H(x^*)$ is **positive semidefinite**, i.e.,*

$$s^T H(x^*) s \geq 0, \forall s \in \mathbb{R}^n.$$

Proof. If x^* is local minimizer, from the first order necessary condition above, we know $\nabla f(x^*) = 0$. Suppose there exist a direction s such that $s^T H(x^*) s < 0$, then the Taylor expansion along s will give

$$f(x^* + as) = f(x^*) + \frac{1}{2} a^2 s^T H(x^*) s + O(a^3).$$

We select a sufficiently small $a > 0$ such that

$$\frac{1}{2} a^2 s^T H(x^*) s + O(a^3) < 0$$

Then, we have

$$f(x^* + as) - f(x^*) < 0,$$

which contradicts with the fact that $f(x^*)$ is local minimum. □

Theorem 7.1.3 (second order sufficient optimality condition, strict local minimizer). *[1] Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, if $\nabla f(x^*) = 0$ and $H = \nabla^2 f(x^*)$ is **positive definite**, then x^* is local minimizer.*

In fact, if the above conditions holds, it can be showed that there exists scalars $\gamma > 0$ and $\epsilon > 0$ such that

$$f(x) \geq f(x^*) + \frac{\gamma}{2} \|x - x^*\|, \forall x \text{ such that } \|x - x^*\| < \epsilon.$$

Proof. Consider an open Ball $B(x^*, \epsilon)$ for some ϵ. For any $s \neq 0, x^* + s \in B(x^*, \epsilon)$, we have

$$\begin{aligned} f(x^* + s) &= f(x^*) + g(x^*)^T s + \frac{1}{2} s^T H(x^*) s + o(\|s\|^2) \\ &\geq f(x^*) + \frac{1}{2} \lambda \|s\|^2 + o(\|s\|^2) \\ &= f(x^*) + \frac{1}{2} \|s\|^2 \left(\lambda + \frac{o(\|s\|^2)}{\|s\|^2}\right) \\ &> f(x^*), \end{aligned}$$

for sufficiently small ϵ. Note that $\lambda > 0$ is the smallest eigenvalue of $H(x^*)$, and we use Rayleigh quotient [Theorem 5.8.4]that

$$s^T Hs \geq \lambda \|s\|^2.$$

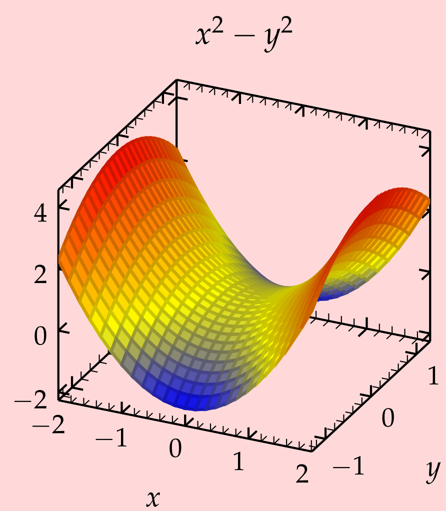
□

Remark 7.1.3 (lack of first-order sufficient condition). Note that **there is no first-order sufficient condition**, i.e., any first-order condition cannot guarantee sufficiency unless we know the function is convex [Theorem 10.5.3].

Example 7.1.2.

- $f(x) = x^2, f'(x) = 2x = 0, x^* = 0, f''(x^*) = 2 > 0$. Therefore, $x = 0$ is a local minimum.
- $f(x) = x^3, f'(x) = 3x^2 = 0, x^* = 0, f''(x^*) = 0$. Therefore, $x = 0$ is not a local minimum or maximum.
- $f(x) = x^4, f'(x) = 4x^3 = 0, x^* = 0, f''(x^*) = 0$. Note that optimal conditions we cover so far can be determine if $x = 0$ is optimal or not. We need to look for higher order conditions.

Example 7.1.3. Consider the function $f(x, y) = x^2 - y^2$.



Note that at $(0,0)$, gradient is zero, but its hessian $H = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$ is not semi-positive definite. Therefore $(0,0)$ is not a local minimum.

7.1.3 Special case: unconstrained quadratic programming

Theorem 7.1.4 (optimality for unconstrained quadratic programming). *Let $f(x) = c^T x + \frac{1}{2}x^T Hx$, where H is symmetric and $c \neq 0$.*

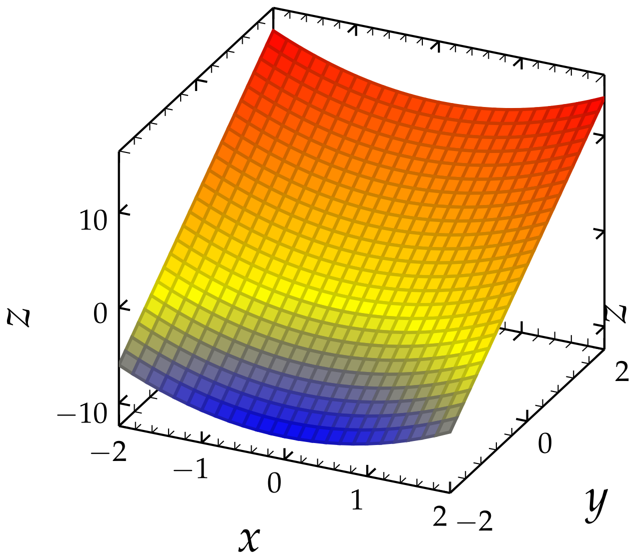
We have the following [also see Figure 7.1.3]

- *$f(x)$ is unbounded below if any the three conditions holds*
  - *$H = 0$.*
  - *$c \notin \mathcal{R}(H)$; that is, equation $Hx^* = -c$ has no solution.*
  - *H has negative eigenvalues.*
- *$f(x)$ has a unique global minimizer $x*$ if $Hx^* = -c$ has a unique solution and H is positive definite.*
- *$f(x)$ has an infinitely many global minimizers $x*$ if $Hx^* = c$ has infinitely many solutions and H is positive semidefinite.*

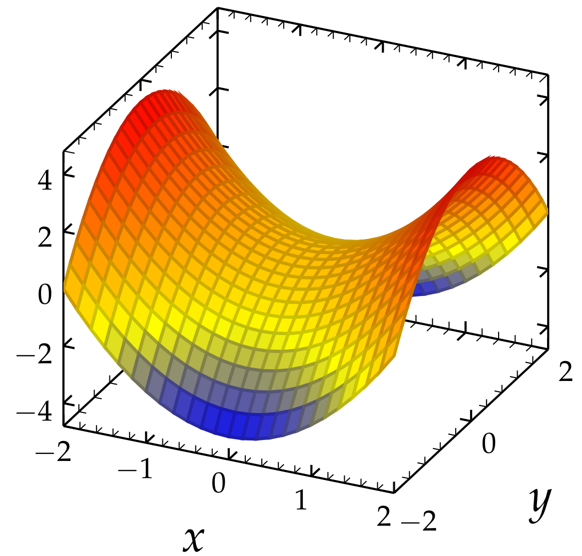
Proof. (1) (a) straight forward. (b) If $Hx^* = c$ has no solution, then c must has component c_N lying in the null space of H. Move along c_N and $f(x)$ can reach ∞. (c) Move along the direction associated with negative eigenvalue and $f(x)$ can reach ∞. (2) From (3) Consider a perturbed point $x^* + p$, we have

$$\begin{aligned}
&\frac{1}{2}(x^* + p)^T H(x^* + p) + c^T(x^* + p) - \frac{1}{2}[x^*]^T Hx^* - c^T x^* \\
=&\frac{1}{2}p^T Hp + c^T p + p^T Hx \\
=&\frac{1}{2}p^T Hp \geq 0
\end{aligned}$$

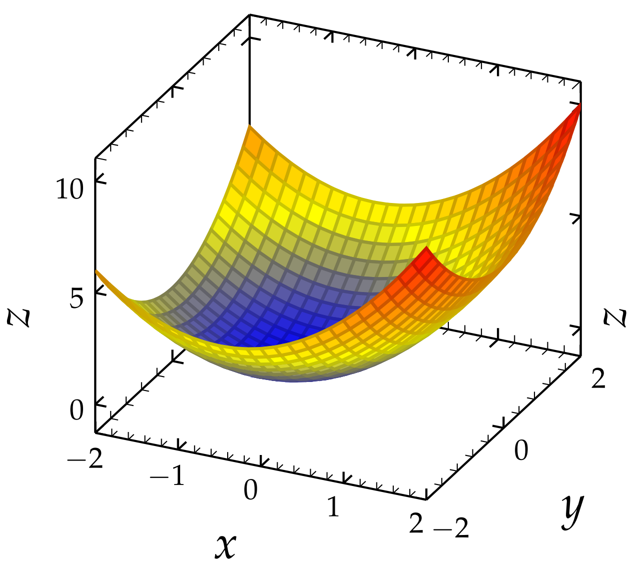
In fact, if $p \in \mathcal{N}(H)$, then $x^* + p$ is also a global minimizer. $\square$



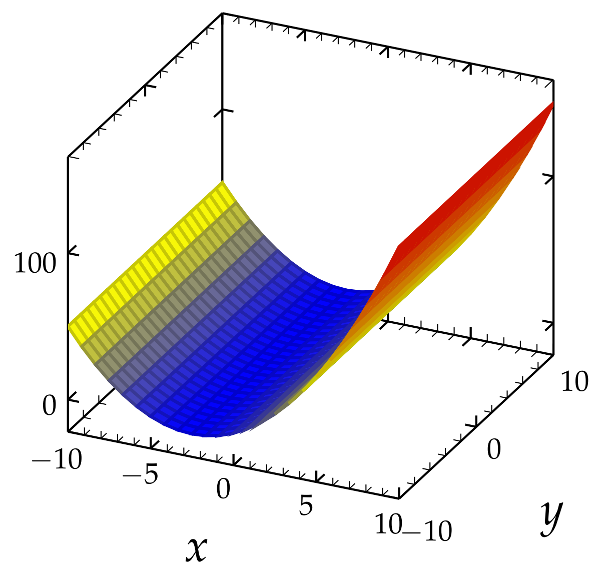
(a) $f(x,y) = 5y + x^2$ where $Hx^* = c$ does not have a solution.



(b) $f(x,y) = x^2 - y^2$ where H has a negative eigenvalue.



(c) $f(x,y) = x^2 + y^2 + x$ where H is positive definite.



(d) $f(x,y) = 5x + x^2$ where H is positive semi-definite and $Hx^* = -c$ has infinitely many solutions.

Figure 7.1.3: Illustration of different cases in unconstrained quadratic optimization.

7.2 Line search method

common notations

- x_k: kth iteration point of x.
- g_k: gradient $\nabla f(x_k)$ at x_k.
- H_k: Hessian $\nabla^2 f(x_k)$ at x_k.

7.2.1 A generic algorithm

Given a differentiable function $f(x)$ and its non-zero gradient[1] $g(x) \triangle \nabla f(x)$, if we follow the direction $-g$ move a sufficiently-small step size α, we can decrease $f(x)$, i.e.,

$$f(x - \alpha g(x)) < f(x)$$

using Taylor expansion.

The direction $-g(x)$ is the steepest descent direction. More generally, we can select a descent direction of similar nature, perform a suitable step size and decrease the $f(x)$. If perform aforementioned steps iteratively, we could constantly decrease $f(x)$ until a minimum is reached.

This is indeed the core idea of **line search** framework. In summary, we are given an objective function $f(x) \in C_1$ (sometimes C_2 is required). Starting from an initial iterate x_0, we repeat the following procedure

- **(find a descent direction)**find a descent direction p_k .
- **(choose step length)** compute a scalar step length α_k such that $f(\alpha_k p_k + x_k) < f(x_k)$
- Generate the next iteration via

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

The procedure can also be summarized by the following algorithm 1.

Algorithm 1: A generic line search algorithm

Input: Initial guess x_0

Set $k = 0$ **repeat**

  Find a decent direction p_k at x_k.

  Compute a suitable step length α_k such that $f(\alpha_k p_k + x_k) < f(x_k)$

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

  Set $k = k + 1$

until *terminal condition is satisfied*;

Output: approximate minimizer x_k

1 If gradient is zero, we have possibly arrived the local minimum, see Theorem 7.1.1

In the following sections, we will elaborate the theory and algorithm regarding the computation of descent directions and step size. By combining different descent direction and step size calculation subroutines, we get different line search algorithms.

7.2.2 Theory and computation of descent directions

7.2.2.1 *Gradient descent direction and properties*

Definition 7.2.1 (descent direction). *A vector $p_k \in \mathbb{R}^N$ at x_k is called a descent direction if*

$$g_k^T p_k < 0$$

if $g^T \neq 0$.

Remark 7.2.1. We are not considering the situation where $g_k = 0$, which suggesting local minimum has arrived(and the iteration process should be terminated.).

Lemma 7.2.1. *Let p_k be a descent direction at x_k. Then there exists a $\alpha > 0$ such that*

$$f(x_k - \alpha p_k) - f(x_k) < 0.$$

Proof. We can write the descent direction as the combination of $-g_k$ and a component perpendicular to g_k. That is, we have

$$p_k = -\beta g_k + \gamma g_k^{\perp}, \beta > 0.$$

Using Taylor expansion, we can show

$$f(x_k - \alpha p_k) = f(x_k) - \alpha\beta \|g_k\|^2 + O(\alpha^2).$$

Then for sufficiently small α, we have

$$f(x_k - \alpha\beta g_k) - f(x_k) < 0.$$

□

Lemma 7.2.2 (steepest decent direction and their properties). *Let x_k be the current iterate. Let g_k be the gradient at x_k.*

- *Let g_k be the gradient at x_k, then $p_k = -g_k$ is a descent direction, and is called **steepest-descent direction**.*

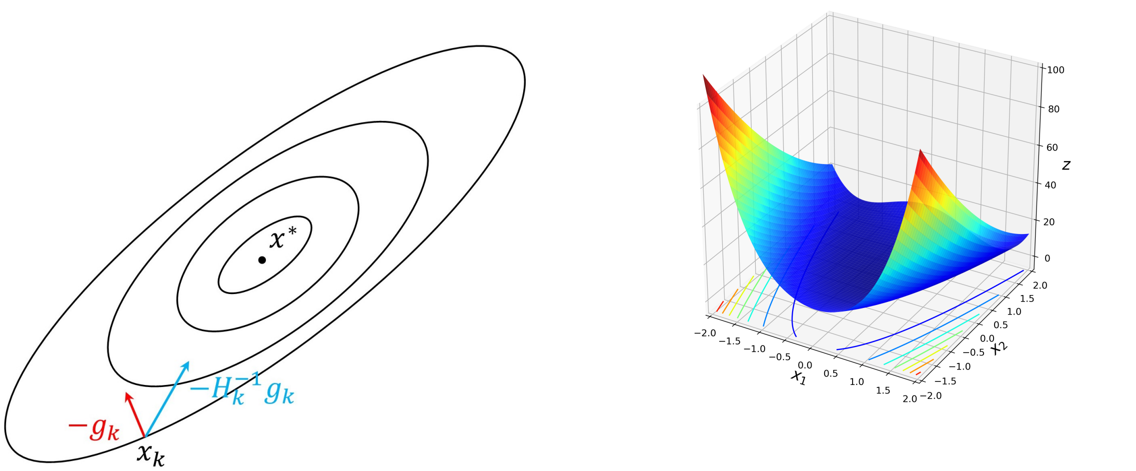
- *The steepest descent direction $p_k = -g_k$ is the solution to a constrained minimization based on a linear model*

$$\min_{p \in \mathbb{R}^n} m_k^L(x_k + p) = f_k + g_k^T p, \text{ subject to } \|p\|_2 = \|g_k\|.$$

- *The steepest descent direction $p_k = -g_k$ is the solution to a strict convex quadratic programming*

$$\min_{p \in \mathbb{R}^n} f_k + g_k^T p + \frac{1}{2} p^T I p.$$

Proof. (1) $g_k^T p_k = -\|g_k\|^2 < 0$ if $g_k \neq 0$. (2) The Lagrangian is $L(p, \lambda) = f_k + g_k^T p + \lambda(\frac{1}{2} p^T p - g_k^T g_k)$. The optimality conditions gives $g_k + \lambda p = 0, \|p\|_2 = \|g_k\|$; that is, $p = -g_k$ or g_k. It is easy to see that minimizer is $-g_k$. (3) Straight forward. □



(a) Comparison of steepest descent direction and Newton step direction.

(b) Steepest gradient descent would perform poorly for the RosenBrock objective function, which contains long and shallow valley.

Figure 7.2.1: Drawbacks of steepest gradient descent.

7.2.2.2 *Curvature-modified descent direction*

Lemma 7.2.3 (modified decent directions). *Let x_k be the current iterate. Let g_k be the gradient at x_k.*

- *Let B_k be a symmetric positive definite matrix, then the p_k is a descent direction if $B_k p_k = -g_k$.*
- *The descent direction p_k as the solution $B_k p_k = -g_k, B_k > 0$ is the solution to a strict convex quadratic programming*

$$\min_{p \in \mathbb{R}^n} f_k + g_k^T p + \frac{1}{2} p^T B_k p.$$

Proof. (1) $g_k^T p_k = -g_k B_k^{-1} g_k < 0$ if $g_k \neq 0$ since B_k^{-1} is also symmetric positive definite [Theorem 5.7.3]. (2) The optimality condition gives $g_k + B_k p = 0 \implies B_k p_k = -g_k$. □

Definition 7.2.2 (Newton direction). *Let g_k and H_k be the gradient and the* ***Hessian*** *at x_k, then the p_k satisfying $H_k p_k = -g_k$ is called* ***Newton direction****.*

Lemma 7.2.4 (properties of Newton direction).

- *Newton's direction is a* ***descent direction*** *if $H_k > 0$.*
- *Let p be the Newton direction and let g_k be the gradient at x_k. It follows that p is the minimizer of*

$$\min_{p \in \mathbb{R}^n} \frac{1}{2} p^T H_k p + g_k^T x,$$

if $H_k > 0$.

Proof. (1) $g^T p = -g^T H^{-1} g < 0$.(2) Consider the objective function

$$\frac{1}{2} p^T H_k p + g_k^T x.$$

The optimality condition is

$$H_k p + g_k = 0 \implies H_k p = -g_k.$$

That is p is also the Newton direction. □

Example 7.2.1. Consider the optimization

$$\min(x_1, x_2) = (x_1 - 2)^4 + (x_1 - 3x_2)^2 + \exp(2x_1 - 2).$$

The gradient is given by

$$g = \nabla f(x_1, x_2) = \begin{bmatrix} 4(x_1 - 2)^3 + 2(x_1 - 3x_2) + 2\exp(2x_1 - 2) \\ -4(x_1 - 3x_2) \end{bmatrix},$$

and Hessian is given by

$$H = \nabla^2 f(x_1, x_2) = \begin{bmatrix} 12(x_1 - 2)^2 + 2 + 4\exp(2x_1 - 2) & -6 \\ -6 & 18 \end{bmatrix}.$$

The Newton direction can be evaluated by

$$p_{NW} = -H^{-1} \cdot g.$$

If the Hessian H_k is not positive definite (H_k will always be symmetric), then the Newton direction is not a descent direction. In the following, we provide a way to generate a positive definite matrix B_k from H_k, that also at the same time contains some curative information.

Methodology 7.2.1 (Hessian modification method via eigenvalue spectrum). *Given a symmetric matrix H with decomposition $H = V\Lambda V^T$, we can use the following procedures to generate a positive definite matrix B:*

- *Make all eigenvalue $\lambda_i' = \epsilon > 0$ if $\lambda_i < \epsilon$; otherwise $\lambda_i' = \lambda_i$. And the modified matrix is $B = V\Lambda' V^T$.*
- *Make all eigenvalue $\lambda_i' = -\lambda_i$ if $\lambda_i < -\epsilon$; $\lambda_i' = \lambda_i$ if $\lambda_i' > \epsilon$; $\lambda_i' = \epsilon$ otherwise. And the modified matrix is $H' = V\Lambda' V^T$. And the modified matrix is $B = V\Lambda' V^T$.*
- *Directly generate $B = H + (\min\{\lambda_i\} + \epsilon)I$.*

7.2.2.3 *Quasi-Newton method*

In Methodology 7.2.1, we introduced a rather Naive way to modify the spectrum Hessian H, which contains curative information, to a positive definite matrix B, in order to generate descent direction p satisfying $Bp = -g$.

This section, we discuss an alternative way, known as Quasi-Newton method, to generate B that approximate H. In the Quasi-Newton method, we aims to efficiently compute B_{k+1} that satisfies

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

which is known as **Secant equation**. Intuitively, Secant equation is aiming to solve Hessian via finite different. If x is one-dimension, we recover the classical root finding Secant method [Methodology A.14.3].

Denote $s_k = (x_{k+1} - x_k)$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$,the secant equation is also written as $B_{k+1}s_k = y_k$.

Note that Secant equation is an underdetermined system for B. There are many Quasi-Newton algorithms [2, p. 135], which give different ways to compute B that satisfies Secant equation.

Here, we cover the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, which solve B_{k+1} on top of B_k via

$$B_{k+1} = B_k + \alpha uu^T + \beta vv^T$$

with $u = y_k$ and $v = B_k s_k$

By requiring the satisfactions of the Secant equation, we have

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

Using matrix inversion formula [Lemma A.8.3], we can get

$$B_{k+1}^{-1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) B_k^{-1} \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}.$$

The BFGS gives a positive definite B_{k+1} if certain condition is satisfied, as given in the following.

Lemma 7.2.5 (Positive definiteness of BFFS update). *Let B_k be positive definite. Then*

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

is posiitive definite if $s_k^T y_k > 0$.

Proof.

$$\begin{aligned} x^T B_{k+1} x &= x^T B_k x + \frac{\left(s_k^T x\right)^2}{s_k^T y_k} - \frac{\left(y_k^T B_k x\right)^2}{y_k^T B_k y_k} \\ &= \frac{y_k^T B_k y_k x^T B_k x - \left(y_k^T B_k x\right)^2}{y_k^T B_k y_k} + \frac{\left(s_k^T x\right)^2}{s_k^T y_k} \end{aligned}$$

If one defines the dot product $\langle x, y \rangle$ as $x^T B_k y$, the above equation reads

$$x^T B_{k+1} x = \frac{\langle y_k, y_k \rangle \langle x, x \rangle - \langle y_k, x \rangle^2}{\langle y_k, y_k \rangle} + \frac{\left(s_k^T x\right)^2}{s_k^T y_k}$$

where the first term is positive due to Cauchy-Schwartz inequality, and the second is positive since $s_k^T y_k > 0$. □

As we will see later, the condition $s_k^T y_k > 0$ is satisfied if the step size is chosen according to the Wolfe condition [Definition 7.2.4].

7.2.2.4 *Subspace optimization in quadratic forms*

In this section, we study a simple quadratic optimization problem

$$f(x) = \frac{1}{2} x^T H x.$$

We are particularly interested in the relationship between minimization and eigenvalue spectrum. Intuitively, if we want to increase the objective function, we can move away from origin along the positive eigenvector direction. Or we can move closer to the origin along the negative eigenvector direction.

We present the results in the following Lemma.

Lemma 7.2.6. *Consider objective function*

$$f(x) = \frac{1}{2} x^T H x$$

where $x \in \mathbb{R}^N$, and $H \in \mathbb{R}^{N \times N}$ is a symmetric and nonsingular matrix. Let the spectral decomposition of H be $H = V \Lambda V^T$ Therefore, if we want to increase $f(x)$ from an initial point x_0,

- *we can move in the direction $p = V_i (V_i^T x)$ where $\lambda_i > 0$; that is $f(x_0 + \alpha p) > f(x_0)$, if $\alpha > 0$ is sufficiently small.*
- *we can move in the direction $p = -V_j (V_j^T x)$ where $\lambda_j < 0$; that is $f(x_0 + \alpha p) > f(x_0)$, if $\alpha > 0$ is sufficiently small.*

Similarly, if we want to decrease $f(x)$,

- *we can move in the direction $p = -V_i (V_i^T x)$ where $\lambda_i > 0$; that is $f(x_0 + \alpha p) < f(x_0)$, if $\alpha > 0$ is sufficiently small..*

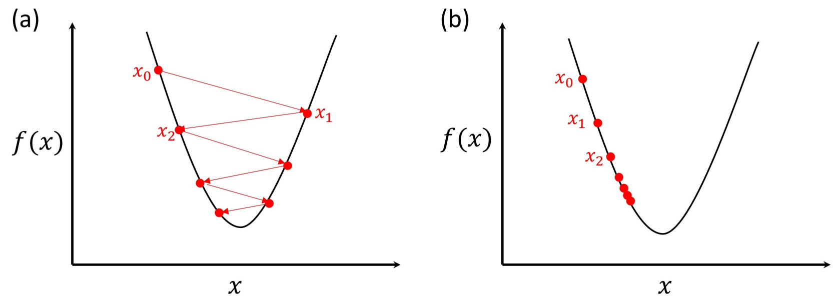


Figure 7.2.2: Demonstration on the step choices on the iterative algorithm. (a) Large step size. (b) Small step size.

- *we can move in the direction* $p = V_j(V_j^T x)$ *where* $\lambda_j < 0$*; that is* $f(x_0 + \alpha p) < f(x_0)$*, if* $\alpha > 0$ *is sufficiently small.*

Proof. $f(x) = \frac{1}{2}x^T H x = (V^T x)^T \Lambda (V^T x) = \sum_{i=1}^N \lambda_i \frac{1}{2}(V_i^T x)^2$. Then, we can show $\nabla f = \sum_{i=1}^N \lambda_i (V_i^T x) V_i$. And for $\lambda_i > 0$,$(V_i^T x)V_i$ will be the ascent direction. Similar arguments for $\lambda_i < 0$ cases. □

7.2.3 Theory and computation of step length

7.2.3.1 *Overview*

After determining the descent direction to update iterate, we need to determine a suitable step size α to update iterate $x_k = x_{k-1} + \alpha p_k$. The step size α has to be chosen with caution. As shown in Figure 7.2.2, a large step size can result in oscillation of iterates around the local minimum and thus slow down the convergence or even cause divergence. On the other hand, a sufficiently small step size can require extensive iteration to converge, or even fail to converge.

In the following, we first consider some theories for choosing step sizes for well-behaved and relatively simple objective functions. Then we move a more general numerical procedures to determine the step size at each iteration.

7.2.3.2 *Lipschitz bounded convex functions*

Theorem 7.2.1. *Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and additionally*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| for\ any\ x, y$$

i.e. , ∇f is Lipschitz continuous with constant $L > 0$

Gradient descent with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)} - f(x^\star) \leq \frac{\|x^{(0)} - x^\star\|^2}{2tk}$$

i.e. gradient descent has convergence rate $O(1/k)$
i.e. to get $f(x^{(k)}) - f(x^\star) \leq \epsilon$, we need $O(1/\epsilon)$ iterations

Proof. Since ∇f Lipschitz with constant L, which means $\nabla^2 f \preceq LI$, we have $\forall x, y, z$

$$(x - y)^T(\nabla^2 f(z) - LI)(x - y) \leq 0$$

Which means

$$L\|x - y\|^2 \geq (x - y)^T \nabla^2 f(z)(x - y)$$

Based on Taylor's Remainder Theorem, we have $\forall x, y$, $\exists z \in [x, y]$

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}(x - y)^T \nabla^2 f(z)(x - y) \\ &\leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}\|y - x\|^2 \end{aligned} \tag{1}$$

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

$$\begin{aligned} f(x^+) &\leq f(x) + \nabla f(x)^T(x - t\nabla x - x) + \frac{L}{2}\|x - t\nabla x - x\|^2 \\ &= f(x) - (1 - \frac{Lt}{2})t\|\nabla f(x)\|^2 \end{aligned} \tag{2}$$

Taking $0 < t \leq 1/L$, $1 - Lt/2 \geq 1/2$, we have

$$f(x^+) \leq f(x) - \frac{t}{2}\|\nabla f(x)\|^2$$

Since f is convex, $f(x) \leq f(x^*) + \nabla f(x)^T (x - x^*)$ we have

$$\begin{aligned} f(x^+) &\leq f(x) - \frac{t}{2}\|\nabla f(x)\|^2 \\ &\leq f(x^*) + \nabla f(x)^T(x - x^*) - \frac{t}{2}\|\nabla f(x)\|^2 \\ &= f(x^*) + \frac{1}{2t}(\|x - x^*\|^2 - \|x - x^* - t\nabla f(x)\|^2) \\ &= f(x^*) + \frac{1}{2t}(\|x - x^*\|^2 - \|x^+ - x^*\|^2) \end{aligned} \tag{3}$$

Summing over iterations, we have

$$\begin{aligned} \sum_{i=1}^{k}(f(x^{(i)} - f(x^*)) &\leq \frac{1}{2t}(\|x^{(0)} - x^*\|^2 - \|x^{(k)} - x^*\|^2) \\ &\leq \frac{1}{2t}\|x^{(0)} - x^*\|^2 \end{aligned} \tag{4}$$

From (2), we can see that $f(x^{(k)})$ is nonincreasing. Then we have

$$f(x^{(k)}) - f(x^*) \leq \frac{1}{k}\sum_{i=1}^{k}(f(x^{(i)} - f(x^*)) \leq \frac{\|x^{(0)} - x^*\|^2}{2tk}$$

□

Lemma 7.2.7 (theoretical optimal step size for unconstrained quadratic optimization). *Consider a unconstrained quadratic minimization problem*

$$\min_{x\in\mathbb{R}^n} f(x) \triangleq \frac{1}{2}x^T H x + b^T x,$$

where $H \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n$. Let $\lambda_1 \geq \lambda_2 ... \geq \lambda_n > 0$ be the eigenvalues of H. It follows that

- *$x_{k+1} = x_k - \alpha\nabla f(x_k)$ can be written as $(x_{k+1} - x^*) = (I - \alpha H)(x_k - x^*)$, where x^* is the unique minimizer of $f(x)$.*
- *If step size $0 < \alpha < 2/\|H\|_2$, then the operator $\|I - \alpha H\|_2 < 1$; that is, $I - \alpha H$ is a contraction.*
- *If $\alpha = 2/(\lambda_1 + \lambda_n)$, then $I - \alpha H$ has the minimum 2-norm.*

Proof. (1) Note that $\nabla f(x_k) = Hx_k - b$ and $Hx^* = b$, therefore

$$x_{k+1} = x_k - \alpha(Hx_k - Hx^*) \Leftrightarrow (x_{k+1} - x^*) = (I - \alpha H)(x_k - x^*)$$

. (2) Note that
$$(x_{k+1} - x^*) = (I - \alpha H)(x_k - x^*)$$
implies [Theorem 5.13.1]
$$\left\|x_{k+1} - x^*\right\| \leq \|I - \alpha H\| \left\|x_k - x^*\right\|.$$
Moreover $\|I - \alpha H\|_2$ equals the maximum **absolute** eigenvalue of $I - \alpha H$ [Theorem 5.9.3], which is $\max|1 - \alpha\lambda_i|$. To let $\|I - \alpha H\|_2 < 1$, we have $\alpha < 2/\lambda_1 = 2/\|H\|_2$. (3) To make $\max|1 - \alpha\lambda_i|$ minimal, we simplify to
$$\min_\alpha \max(|1 - \alpha\lambda_1|, |1 - \alpha\lambda_n|)$$
due to the factor that λ_i are monotone. Then, the minimum value is reached at
$$1 - \alpha\lambda_1 = -(1 - \alpha\lambda_n).$$

□

Remark 7.2.2 (convergence rate). At optimal choice of α, we have
$$\|I - \alpha H\|_2 = \frac{\lambda_1/\lambda_n - 1}{\lambda_1/\lambda_n + 1} = \frac{cond(H) - 1}{cond(H) + 1}.$$
We have the following statement about convergence rate (using contraction theory section 6.2]:

- 'fast linear convergence' when $cond(H) \approx 1$.
- 'slow linear convergence' when $cond(H) \gg 1$.
- 'convergence in a single iteration' when $cond(H) = 1$, i.e., for diagonal matrices.

Remark 7.2.3 (compare with conjugate gradient descent). For this type of problem, conjugate gradient descent at most takes n iteration; but using our steepest descent method can take much more longer.

7.2.3.3 *Backtracking-Armijo step side search*

One way to choose step size is to require that the new iterate can lead to an improvement in the function value is at least a fraction of the improvement predicted by the linear approximation [Figure 7.2.3]. Mathematically, the condition is known as **Armijo sufficient decrease condition**, defined by

Definition 7.2.3 (Armijo sufficient decrease condition). *Given a point x_k and a search direction p_k, we say the step size α_k satisfies the **Armijo condition** if*

$$f(x_k + \alpha_k p_k) \le f(x_k) + \eta \alpha_k g_k^T p_k$$

for some $\eta \in (0, 1)$.

To ensure the decrease in f is substantial enough to ensure convergence. When we choose a small η, we tend to select large α_k to ensure the condition to hold.

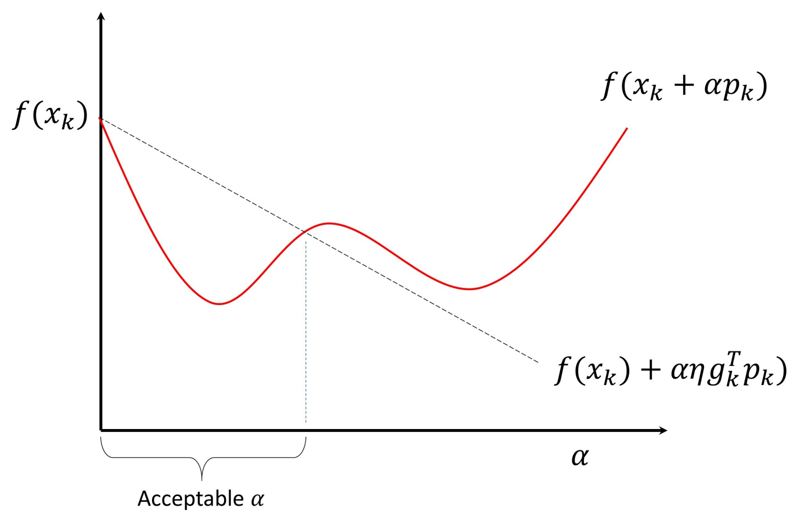


Figure 7.2.3: Armijo sufficient decrease condition.

Algorithm 2: Backtracking-Armijo line search algorithm

Input: Initial guess x_k, p_k

1 Choose $\alpha_{init} > 0, \eta \in (0, 1)$, and $\tau \in (0, 1)$
2 Set $\alpha_0 = \alpha_{init}$ and $l = 0$
3 Set $l = 0$ **repeat**
4   Set $\alpha_{l+1} = \tau \alpha_l$
5   Set $l = l + 1$
6 **until** *terminal condition* $f(x_k + \alpha_l p_k) \le f(x_k) + \eta \alpha_l g_k^T p_k$ *is satisfied;*

Output: approximate minimizer x_k

Lemma 7.2.8 (existence of step length satisfying Armijo condition will always satisfied in). *Suppose that*

- $f \in C^1$ *and* $g(x)$ *is Lipschitz continuous with Lipschitz constant* $\gamma(x)$
- p *is a descent direction at* x

Then for any given $\eta \in (0,1)$, *the Armijo condition*

$$f(x+\alpha p) \leq f(x) + \eta\alpha g(x)^T p$$

is satisfied for all $\alpha \in [0, \alpha_{max}]$, *where*

$$\alpha_{max} = \frac{2(\eta-1)g(x)^T p}{\gamma(x)\|p\|_2^2} > 0.$$

Proof.

$$\begin{aligned} f(x+\alpha p) &\leq f(x) + \alpha g(x)^T + \frac{1}{2}\gamma(x)\alpha^2\|p\|_2^2 \\ &\leq f(x) + \alpha g(x)^T p + \alpha(\eta-1)g(x)^T p \\ &= f(x) + \alpha\eta g(x)^T p \end{aligned}$$

where we use the fact that

$$\alpha \leq \frac{2(\eta-1)g(x)^T p}{\gamma(x)\|p\|_2^2}.$$

□

Remark 7.2.4 (backtracking algorithm will terminate in finite steps). Because the backtracking algorithm is always shrinking the step-length, it will eventually terminated with an α falling inside $[0, \alpha_{max}]$.

7.2.3.4 *Wolfe condition*

A more complex step size condition is the Wolfe condition, which requries both sufficient decrement of objective value and the slope.

Definition 7.2.4 (Wolfe conditions). *Given the current iterate* x_k, *search direction* p_k, *and constants* $0 < c_1 < c_2 < 1$, *we say that the step length* α_k *satisfies the* ***Wolfe conditions*** *if*

$$\begin{aligned} f\left(x_k + \alpha_k p_k\right) &\leq f\left(x_k\right) + c_1\alpha_k \nabla f\left(x_k\right)^T p_k \\ \nabla f\left(x_k + \alpha_k p_k\right)^T p_k &\geq c_2 \nabla f\left(x_k\right)^T p_k \end{aligned}$$

Lemma 7.2.9 (Wolfe conditions for positive definite matrix in BFGS update). *Consider BFGS update where $B_k > 0$ and p_k is a descent direction for f at x_k. If α_k satisfies Wolfe condition such that $x_{k+1} = x_k + \alpha_k p_k$, then*

$$y_k^T s_k > 0,$$

where $s_k = (x_{k+1} - x_k)$ and $y_k = g_{k+1} - g_k$.

With such condition satisfied, $B_{k+1} > 0$.

Proof. From the Wolfe condition, we have

$$\nabla f\left(x_k + \alpha_k p_k\right)^T p_k \geq c_2 \nabla f\left(x_k\right)^T p_k.$$

Multiplying both sides by α_k and using $x_{k+1} = x_k + \alpha_k p_k$, we have

$$g_{k+1}^T s_k \geq c_2 g_k^T s_k,$$

and

$$g_{k+1}^T s_k - g_k^T s_k \geq c_2 g_k^T s_k - g_k^T s_k.$$

We then have

$$y_k^T s_k \geq \left(c_2 - 1\right) g_k^T s_k = \alpha_k \left(c_2 - 1\right) g_k^T p_k > 0$$

due to the fact hat $c_2 \in (0, 1)$ and $g_k^T p_k < 0$ (p_k is a descent direction). □

7.2.4 Complete algorithms

In the following, we give three complete algorithms that combine descent direction and step size computations. Note that Quasi-Newton method has to be used to Wolfe condition to guarantee positive definiteness of B_{k+1}. A comprehensive convergence analysis can be found in [2].

Algorithm 3: Steepest decent Backtracking-Armijo line search algorithm

Input: Initial guess x_0

Set $k = 0$.

repeat

  Compute a steepest decent direction $p_k = -g_k$ at x_k.

  Compute a suitable step length α_k use Backtracking-Armijo line search algorithm

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

  Set $k = k + 1$

until $\|\nabla f(x_k)\| \leq 10^{-8} \max(1, \|\nabla f(x_0)\|)$;

Output: approximate minimizer x_k

Algorithm 4: Modified Newton Backtracking-Armijo line search algorithm

Input: Initial guess x_0

Set $k = 0$.

repeat

  Set a decent direction $p_k = -B_k^{-1} g_k$ at x_k, where B_k is a symmetric positive definite matrix modified from Hessian H_k [Methodology 7.2.1].

  Compute a suitable step length α_k use Backtracking-Armijo line search algorithm

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

  Set $k = k + 1$

until $\|\nabla f(x_k)\| \leq 10^{-8} \max(1, \|\nabla f(x_0)\|)$;

Output: approximate minimizer x_k

Algorithm 5: Quasi Newton with Wolfe line search algorithm

Input: Initial guess x_0, and inital positive definite B_0 approximate to the Hessian.

Set $k = 0$.

repeat

  Get a decent direction $p_k = -B_k^{-1} g_k$ at x_k.

  Compute a suitable step length α_k satisfies Wolfe condition.

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

  Update

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

  where $s_k = (x_{k+1} - x_k)$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

  Set $k = k + 1$

until $\|\nabla f(x_k)\| \leq 10^{-8} \max(1, \|\nabla f(x_0)\|)$;

Output: approximate minimizer x_k

7.3 Trust region method

7.3.1 Motivation and the framework

In each iteration, line search methods uses a proxy model to generate a descent direction, and determine a suitable step size that would decrease the objective function sufficiently along this direction. Trust region methods, in some ways, are setting a step size limit (determined by the so-called **trust region**) first, then find a minimizer from a proxy model that aims to decrease the value of the objective function. In Trust region methods, the proxy problem to be solve is defined as follows.

Definition 7.3.1 (trust-region subproblem). *The trust-region subproblem at kth iterate is*

$$\min_{x\in\mathbb{R}^n} m_k(s) = f_k + g_k^T s + \frac{1}{2}s^T B_k s, \text{ subject to } \|s\| \leq \delta_k$$

where $f_k = f(x_k), g_k = \nabla f(x_k)$, B_k *is a symmetric matrix and* $\delta_k > 0$ *is the* ***trust region radius****, and the norm is 2-norm.*

In trust region methods, in general we want B_k to approximate the Hessian $\nabla^2 f$ even the Hessian is not positive definite. In line search methods, we need B_k to be positive definite to generate descent directions.

The size of the trust region δ_k plays a critical role in the efficiency of algorithm since it bounds how far the iterate can move in each step. If chosen too small, the iterate cannot move too much each step although the proxy model well approximates the original objective function. If chosen too large, the proxy model might be a poor approximate to the original objective function, and a minimizer from a poor proxy model would not help find the true minimizer.

We measure the performance of current step by the reduction ratio of actual objective function reduction over the proxy model reduction.

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$$

The value of ρ_k provides important information that we could leverage to adaptively adjust the trust region radius. There are following situations.

- If ρ_k is negative or small then a threshold η_1, then $f(x_k + p_k) > f(x_k)$, which is undesirable, and we reject current step and re-search a new minimizer in a smaller range in next iteration.

- If ρ_k is greater than a threshold $\eta_2, \eta 1 < \eta_2 < 1$, then the proxy model excellently agrees with the true objective function, and we can expand the search range in the next iteration.
- If ρ_k is positive but fall below η_1, then the proxy model reasonably agrees with the true objective function, and we keep the current search range in the next iteration.

A generic trust-region algorithm is given by algorithm 6. An intuition on the convergence of the algorithm is: At sufficiently small δ, quadratic proxy model m is always a good proxy model to f. As the iterate continues to decrease the value of m, eventually a local minimum will be reached.

Algorithm 6: A generic trust-region algorithm

Input: Initial guess x_0

Choose $\delta_0 > 0, 0 < \gamma_d < 1 < \gamma_i$, and $0 < \eta_1 \leq \eta_2 < 1$

Set $k = 0$

repeat

  Compute (exactly or approximately) a search direction p_k as the solution to

$$\min_{x\in\mathbb{R}^n} m_k(s) = f_k + g_k^T s + \frac{1}{2} s^T B_k s, \text{ subject to } \|s\| \leq \delta_k.$$

  Set $\rho_k = \frac{f(x_k) - f(x_k + s_k)}{\Delta m_k(s_k)}$

  if $\rho_k \geq \eta_2$ **then**

    set $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \gamma_i \delta_k$. (very successful)

  if $\rho_k > \eta_1$ **then**

    set $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \delta_k$ (successful)

  else

    set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma_d \delta_k$. (unsuccessful)

  end

  Set $k = k + 1$

until $\|\nabla f(x_k)\| \leq 10^{-8} \max(1, \|\nabla f(x_0)\|)$;

Output: approximate minimizer x_k

7.3.2 Cauchy point method

The Cauchy point method is to solve an minimizer of the optimization problem

$$\min_{x\in\mathbb{R}^n} m_k(s) = f_k + g_k^T s + \frac{1}{2} s^T B_k s, \text{ subject to } \|s\| \leq \delta_k$$

along the steepest descent direction $-g_k$.

More formally, we can define the following Cauchy point subproblem.

Definition 7.3.2 (Cauchy point of the trust-region subproblem). *The Cauchy point method is to solve the subproblem given by*

$$\min_{\alpha \geq 0} m_k(-\alpha g_k), \text{ subject to } \alpha \|g_k\| \leq \delta_k,$$

where $m_k(x) = f_k + g_k^T x + \frac{1}{2} x^T B_k x$.

The solution α_k^C *gives the* ***Cauchy point***:

$$s_k^C = -\alpha_k^C g_k.$$

Lemma 7.3.1 (Cauchy point solution). *Let* $m_k(s)$ *be a trust-region subproblem and let* s_k^C *be the Cauchy point. Further denote* $\Delta m_k(s) = m_k(0) - m_k(s)$*, then*

$$\alpha_k = \begin{cases} \frac{\delta_k}{\|g_k\|} & \text{if } g_k^T B_k g_k \leq 0 \\ \min\left(\|g_k\|^2 / \left(g_k^T B_k g_k \right), \frac{\delta_k}{\|g_k\|} \right) & \text{otherwise} \end{cases}$$

Proof. Note that

$$m_k(-\alpha g_k) = f_k - \alpha \|g_k\|_2^2 + \frac{1}{2} \alpha^2 g_k^T B_k g_k.$$

If $g_k^T B_k g_k < 0$, m_k will continue to decrease as we increase α. Therefore the minimum will be found at the trust-region boundary; that is

$$a_k^C = \frac{\delta_k}{\|g_k\|}.$$

If $g_k^T B_k g_k < 0$, the minimum of m_k will be found either inside the trust region or at the trust-region boundary. Let's first assume the constraint does not exist, then the minimum is obtained by solving a one-dimensional quadratic optimization problem, which has solution given by

$$a_k^* = \frac{\|g_k\|^2}{g_k^T B_k g_k}.$$

If $a_k^* \|g_k\| \geq \delta_k$, the Cauchy point will lie on the boundary. We have

$$a_k^C = \frac{\delta_k}{\|g_k\|}.$$

If $a_k^* \|g_k\| < \delta_k$, the Cauchy point will lie within the boundary. We have

$$a_k^C = a_k^* = \frac{\|g_k\|^2}{g_k^T B_k g_k}.$$

□

As we can see, the Cauchy point barely utilize the curvature information on the matrix B_k in calculating the direction; in fact, it is used only to determine step length. Inherently, Cauchy point method is the steepest descent method under the umbrella of trust-region method and is expected to surfer from issues of the steepest descent method.

7.3.3 Exact solution method

In this section, we study the theory used to find out the exact solution of the trust-region subproblem. In practice, exact method is rarely used except for simple low-dimensional problems.

Theorem 7.3.1 (global minimizer condition for trust-region subproblem). *[2, p. 90]* *A vector s^* is a* ***global*** *minimizer of*

$$\min_{s \in \mathbb{R}^n} m(s) = f + s^T g + \frac{1}{2} s^T B s, \text{ subject to } \|s\|_2 \leq \delta$$

if and only if $\|s^*\| \leq \delta$ and there exists a scalar $\lambda^* \geq 0$ such that

- $(B + \lambda^* I)s^* = -g$
- *$B + \lambda^* I$ is positive semi-definite*
- *(complementary slackness)$\lambda^*(\|s\|_2 - \delta) = 0$*

Moreover, if $B + \lambda^* I$ is positive definite, then s^* is unique.

Proof. See the reference for the full proof. More details on a weaker condition can be found in KKT theory [Theorem 8.4.3 and Theorem 8.4.4]. □

7.3.4 Approximate method

Exact methods are usually quite expensive as it often involves another iterative routine to solve the subproblem. The problem is, even with the exact solution, the iterate could get rejected if the trust-region radius is too large. On the other hand, the Cauchy point method, which simply steepest descent, could be inherently slow.

In this section, we introduce approximate methods that lies in between the Cauchy point method and the exact solution.

The dogleg method finds an approximate solution by replacing the curved trajectory for $p(\delta)$ with a path consisting of two line segments

- The first line segment runs from the origin to the minimizer of m along the steepest descent direction, where
$$p^{\mathrm{U}} = -\frac{g^T g}{g^T B g} g.$$
- The second line segment runs from p^U to p^B, where
$$p^B = -B^{-1} g.$$

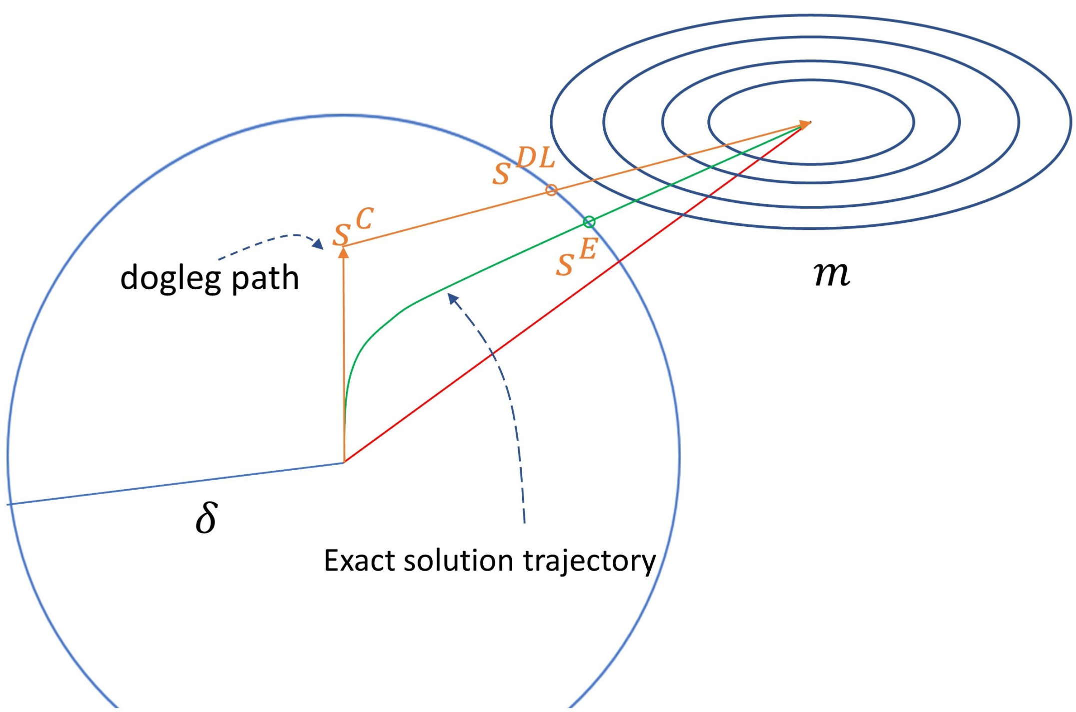


Figure 7.3.1: Demonstration of the dogleg path as an approximation to the exact solution path in the trust-region subproblem.

$$\tilde{p}(\tau) = \begin{cases} \tau p^{\mathrm{U}}, & 0 \leq \tau \leq 1 \\ p^{\mathrm{v}} + (\tau - 1)\left(p^{\mathrm{B}} - p^{\mathrm{U}}\right), & 1 \leq \tau \leq 2 \end{cases}$$

The minimizer alone this path can be found easily [2, p. 80].

If the trust region is large enough, then the minimizer will likely fall on the second segment that contains curvature information; otherwise, the minimizer will fall on the first segment, similar to Cauchy point method.

7.4 Conjugate gradient method

7.4.1 Motivating problems

Definition 7.4.1 (Problem of interest). *Given a symmetric positive-definite matrix A, solve the linear system*

$$Ax = b$$

which is equivalent to finding the unique minimizer of

$$\min_{x \in \mathbb{R}^n} q(x) = \frac{1}{2} x^T A x - b^T x,$$

which has necessary condition $\nabla q(x) = Ax - b = 0$ (due to strong convexity of q and Theorem 10.5.2].

On approach, known as coordinate descent, is to minimize each (orthogonal) dimension one at a time, and hopefully solve a n dimension problems in n steps [Figure 7.4.1(a)]. This is exactly the case when A is diagonal such that the columns of A are colinear with standard vectors, and we minimize along each column vector/coordinate every iteration. When A is not diagonal, coordinate descent can take iterations far than n to converge to the true solution.

On the other hand, we can perform eigendecomposition on A and then optimize along each eigenvector direction. However, this method is unlikely to scale to large systems due to the prohibitive cost of eigendecomposition.

Conjugate gradient method aims to seeks n such special direction to optimize in an efficient way.

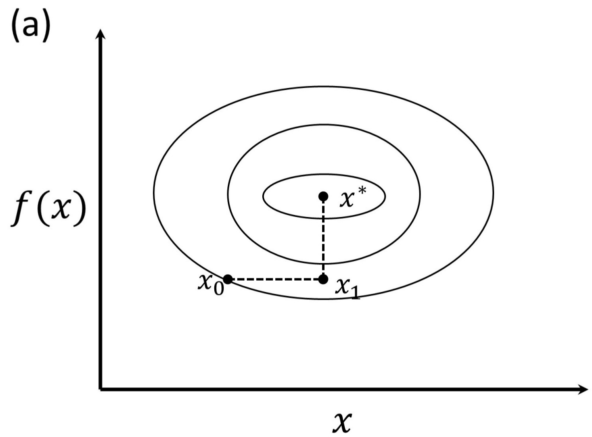


Figure 7.4.1: Demonstration of coordinate descent procedures when A is diagonal and non-diagonal.

7.4.2 Theory conjugate direction

Definition 7.4.2 (A conjugate directions). *A set of nonzero vector $\{s_0, s_1, ..., s_{n-1}\}$ is said to be conjugate with respect to the symmetric positive-definite matrix A if*

$$s_i^T A s_j = 0, \forall i \neq j$$

Remark 7.4.1 (eigenvectors are A conjugate directions, but not reverse).

- Let A be symmetric positive-definite matrix, then we know that eigenvectors corresponding to distinct eigenvalues are orthogonal to each other.

$$v_i A v_j = v_i \lambda_j v_j = 0$$

- For a set of vectors $v_1, v_2, ..., v_n$ are A conjugate directions, then these vectors are eigenvectors.

Lemma 7.4.1 (conjugate directions are linearly independent). *Any set of vectors $\{p_1, ..., p_n\}$ that are A (A is symmetric positive-definite) conjugate directions will be linearly independent set.*

Proof. Suppose they are linear dependent. WLOG, we have $p_1 = \sum_{j=2}^n a_j p_j$. Multiply both sides by $(Ap_1)^T$, we get $p_1^T A p_1 > 0$ on the left side. Then the right hand side equals 0 due to the conjugation. $\square$

Lemma 7.4.2 (expanding subspace minimization). *[2, p. 103][1, p. 121] For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated via conjugate gradient method are expanding subspace minimizers. That is*

$$x_{k+1} = \arg\min_{x \in \mathcal{M}_k} f(x),$$

where

$$\mathcal{M}_k = \{x | x = x_0 + v, v \in span\{p_0, p_1, ..., p_k\}.$$

Moreover, the iteration x_k converges to the unique minimizer x^* at most n steps.

Proof.

$$\frac{\partial f(x_i + \alpha p_i)}{\partial \alpha} = \nabla f(x_{i+1})^T p_i = 0.$$

For $i = 0, ..., k-1$, we have

$$\begin{aligned}\nabla f(x_{k+1})^T p_i &= (Ax_{k+1} - b)^T p_i \\ &= (x_i + \sum_{j=i+1}^{k} \alpha_j p_j)^T A p_i - b^T p_i \\ &= x_i^T A p_i - b^T p_i = \nabla f(x_i)^T p_i = 0.\end{aligned}$$

Therefore,

$$\frac{f(x_{k+1} + \beta_1 p_1 + ... + \beta_k p_k)}{\partial \beta_i}|_{\beta_1 = \beta_2 = ... = 0} = \nabla f(x_{k+1})^T p_i = 0.$$

□

7.4.3 Linear conjugate gradient algorithm

Definition 7.4.3 (conjugate direction generation algorithms). *Given a starting point $x_0 \in \mathbb{R}^n$ and a set of conjugate directions $\{p_0, p_1, ..., p_{n-1}\}$, let us generate the sequence $\{x_k\}$ by setting*

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

where α_k is the one-dimensional minimizer of the quadratic function ϕ along $x_k + \alpha p_k$, given explicitly by

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}, r_k = Ax_k - b.$$

Lemma 7.4.3. *[2, p. 103] For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated via conjugate direction algorithms converges to the solution x^* of the linear system $Ax = b$ in at most n steps.*

Proof. Since the directions $\{p_i\}$ are linearly independent, they must span the whole space $\mathbb{R}^n$. □

Remark 7.4.2 (special cases: conjugate directions are eigenvectors). If we use eigenvector as conjugate directions, then we can see that we are essentially doing sequential optimization on a set of orthonormal basis. The condition of finality [Theorem 6.4.3] guarantees we arrive at the optimum at n steps.

Lemma 7.4.4 (expanding subspace minimization). *[2, p. 103] For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated via conjugate direction algorithms converges to the solution x^* of the linear system $Ax = b$ in at most n steps.*

Algorithm 7: A linear conjugate algorithm

Input: Initial guess x_0

Set $k = 0$, $r_0 = Ax_0 - b$, $p_0 = -r_0$.

repeat

Compute step size

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

Update residual via

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

Generate new conjugate direction

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

Set $k = k + 1$

until r_k *sufficiently small*;

Output: approximate minimizer x_k

7.5 Least square problems

7.5.1 Linear least square theory and algorithm

7.5.1.1 *Linear least square problems*

Definition 7.5.1 (linear least square problem). *Given a $A \in \mathbb{R}^{m\times n}, b \in R^m$, find a x that solves*

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|^2$$

Remark 7.5.1.

1. We usually assume $m \geq n$, then this is a over-determined system(can be consistent or inconsistent); If $m < n$, there will be infinitely solutions(assuming consistence).
2. $f(x) = \frac{1}{2}b^T b - x^T A^T b + \frac{1}{2}x^T A^T Ax$
3. $f(x)$ is convex, since for any matrix A, we always have $A^T A \geq 0$.
4. If A has full column rank, then $A^T A > 0$, indicating f is strictly convex. Note that for any matrix A, we always have $A^T A \geq 0$, when A has full column rank, then Ax is 0 only when x is 0, there fore $(Ax)^T(Ax) > 0, \forall x \neq 0$.
5. $\nabla f = 0 \Rightarrow A^T Ax - A^T b = 0$

Remark 7.5.2 (extreme value property, existence and uniqueness, global vs. local).

- (existence)The minimal value always exists, and $f_{min} \geq 0$.
  - If $Ax = b$ is consistent, then $f_{min} = 0$.
  - If $Ax = b$ is inconsistent, then f_{min} will be the minimum distance of vector b to the subspace spanned by columns of A.
- (uniqueness) uniqueness of the minimizer depends on the rank of A (no matter consistence).
  - If $m < n$, there will be infinitely many minimizers.
  - If $m \geq n$, there will be infinitely many minimizers(when A has non-trivial null spaces) or unique minimizer(when A has full column rank).
- Any local minimizer x^* such that $\nabla f(x^*)$ will be a global minimizer due to convexity.

Remark 7.5.3 (solution methods).

- Direct methods involve directly solving $A^T Ax = A^T b$ using LU, Cholesky, QR, SVD, and linear CG methods under the assumption of $A^T A > 0$. For details, see [3].
- For recursive methods, see subsection 18.5.3.

7.5.1.2 *SVD methods*

Lemma 7.5.1 (minimum error solution via SVD theory, recap). *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Let A have SVD decomposition Theorem 5.9.1 given by*

$$A = [U_1\ U_2]\begin{bmatrix}\Sigma & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T\\ V_2^T\end{bmatrix}.$$

Then the minimizers of

$$\min_{x\in\mathbb{R}^n}\|Ax-b\|_2^2$$

is a set

$$x^* = V_1\Sigma^{-1}U_1^T b + V_2 y.$$

Among the set, the element with the minimum 2-norm length is

$$x_m^* = V_1\Sigma^{-1}U_1^T b.$$

Proof. Lemma 5.1.9. □

Note 7.5.1 (steps). The above method can be executed using the following procedures:

- Compute economic SVD

$$A = U\Sigma V^T = \sum_{i=1}^{n}\sigma_i u_i v_i^T$$

- Form $y = U^T b$
- Form $z = \Sigma^{-1} y$
- $x^* = Vz$. Note that x^* is unique.

Remark 7.5.4. See more discussions from the perspective of linear equations, see subsection 5.1.5.

7.5.1.3 *Extension to L^p norm optimization*

Definition 7.5.2 (L^p norm linear least square problem). *Given a $A \in \mathbb{R}^{m\times n}, b \in R^m$, find a x that solves*

$$\min_x f(x) = \frac{1}{2}\|Ax - b\|_p,$$

where $\|\cdot\|_p$ is the $p-$norm for a vector.

Algorithm 8: Iteratively reweighted least squares for p norm least square

Input: A small threshold number ϵ, a large value w_{big} for weighting zero residuals

Set $k = 0$

Compute the diagonal matrix W with diagonal element being

$$W_{ii} = p(x_i;\beta)(1 - p(x_i;\beta)), i = 1, 2, ..., N.$$

repeat

compute the diagnoal matrix

$$W^k : w_i^k = \left|y_i - x_i^T b^{(k)}\right|^{2-p},$$

if $\left|y_i - x_i^T b^{(k)}\right| < \epsilon$, set $w_i^{(k)} = w_{big}$.

Compute $b^{(k+1)}$ by solving the weighted least squares problem: minimize $(y - Xb)^T W^{(k)}(y - Xb)$.

set $k = k + 1$.

until *stopping criteria* $\left\|b^{(k+1)} - b^{(k)}\right\| \leq \epsilon$ *is met;*

Output: the optimalized value b.

Remark 7.5.5 (interpretation the weight calculation).

- The algorithm is in [4, p. 233].
- We can formulate the pth power of the norm as

$$\|y - Xb\|_p^p = (y - Xb)^T W(y - Xb),$$

where $W = diag(\left|y_1 - x_1^T b\right|^{2-p}, \left|y_2 - x_2^T b\right|^{2-p}, ..., \left|y_n - x_n^T b\right|^{2-p})$.

7.5.2 nonlinear least square problem

Definition 7.5.3 (Nonlinear least square problem). *Given a function $F : \mathbb{R}^n \to \mathbb{R}^m$, the nonlinear least square problem is to find a vector that solves the optimization problem*

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2$$

Remark 7.5.6. **Notes:**

- we will assume $m \geq n$
- $\nabla f = J^T(x)F(x)$, where

$$J(x) = \nabla F(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_1}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{pmatrix}$$

- $\nabla^2 f = J^T J + \sum_{i=1}^m \nabla^2 F_i(x) F_i(x)$
- f is typically nonconvex
- x is the first-order solution if it satisfies

$$\nabla f = J^T(x)F(x) = 0$$

7.5.3 Line search Gauss-Newton method

Definition 7.5.4 (Gauss-Newton subproblem). *The Gauss-Newton subproblem to compute p as a minimizer of*

$$\min_{p\in\mathbb{R}^n} \frac{1}{2}\|F(x) + J(x)p\|_2^2 = \frac{1}{2}\|F(x)\|_2^2 + p^T J(x)^T F(x) + \frac{1}{2} p^T J(x)^T J(x) p,$$

where x is given.

Lemma 7.5.2 (properties of Gauss-Newton subproblem). *For a Gauss-Newton subproblem, if J is full column rank, then*

- *$J^T J$ is positive definite and the Gauss-Newton problem has a unique minimizer.*
- *If $\nabla f(x) = J(x)^T F(x) \neq 0$, then the minimizer p_G satisfy*

$$p_G^T \nabla f(x) < 0;$$

that is, p_G is a descent direction to the original optimization problem.

Proof. (1) When J^TJ is positive definite, the Gauss-Newton problem is a convex optimization problem. (2) $p_G \nabla f(x) = p_G J(x)^T F(x) = -p_G J(x)^T J(x) p_G < 0$, where we use $F(x) = Jp_G$. □

Algorithm 9: Gauss-Newton method for nonlinear least-square algorithm

Input: Initial guess x_0

Set $k = 0$ **repeat**

  Compute a search direction p_k as the solution to

$$\min_{p \in \mathbb{R}^n} \frac{1}{2} \left\| F(x_k) - J(x_k) p \right\|_2^2$$

  Compute a suitable step length α_k use Backtracking-Armijo line search algorithm

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

  Set $k = k + 1$

until $\|\nabla f(x_k)\| \leq 10^{-8} \max(1, \|\nabla f(x_0)\|)$;

Output: approximate minimizer x_k

Remark 7.5.7 (interpretation).

- We require $J(x_k)$ to be full column rank in the whole process such that p_G can be guaranteed to be descent direction.
- We can also use the steepest descent direction $p = -\nabla f(x) = -J(x)^T F(x)$ as the descent direction; however, the Gauss Newton direction p_G is better because it contains curvature information.

7.5.4 Trust region method

Algorithm 10: Levenberg-Marquardt method for nonlinear least-square algorithm

Input: Initial guess x_0

Choose $\delta_0 > 0, 0 < \gamma_d < 1 < \gamma_i$, and $0 < \eta_s \leq \eta_{vs} < 1$ Set $k = 0$ **repeat**

  Compute a search direction p_k as the solution to

$$\min_{p \in \mathbb{R}^n} \frac{1}{2} \|F(x_k) - J(x_k)p\|_2^2$$

  Set $\rho_k = \frac{f(x_k) - f(x_k + s_k)}{\Delta m_k(s_k)}$

  if $\rho_k \geq \eta_{vs}$ **then**

    set $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \gamma_i \delta_k$

  if $\rho_k > \eta_s$ **then**

    set $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \delta_k$

  else

    set $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \gamma_d \delta_k$

  end

  Set $k = k + 1$

until $\|\nabla f(x_k)\| \leq 10^{-8} \max(1, \|\nabla f(x_0)\|)$;

Output: approximate minimizer x_k

Remark 7.5.8 (interpretation).

- The trust region subproblem here is slightly different from general trust region subproblem which directly use Hessian as the quadratic term.
- The trust region Levenberg-Marquardt algorithm is better than the linear search Gauss-Newton algorithm in handling cases where $J(x_k)$ can have dependent columns.

7.5.5 Application: roots for nonlinear equation

Definition 7.5.5 (roots for nonlinear equation problem). *Given a function $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^n$, and assume $\boldsymbol{F}$ is at least continuously differentiable, find a vector $x^* \in \mathbb{R}^n$ such that*

$$\boldsymbol{F}(x^*) = 0$$

Such x^* is called a root.

If x and F are both one-dimensional, we can use following Newton method.

Algorithm 11: Newton method for root finding

Input: Initial guess x_0

Set $k = 0$ **repeat**

  $x_{k+1} = x_k - \frac{f(x_k)}{f'_k}$

  Set $k = k + 1$

until $\|f(x_k)\| \leq 10^{-8}\max(1, \|\nabla f(x_0)\|)$;

Output: approximate root x_k

Remark 7.5.9 (Interpretation of classic Newton's method).

- The new iterate x_{k+1} is generated by setting the first moder $f(x_k + s) = f(x_k) + f'(x_k)s = 0, s = x_{k+1} - x_k$.
- If $\boldsymbol{F}$ is continuously differentiable, then use Newton's method can converge to the root if initial x_0 starts close enough.

Lemma 7.5.3. *Consider the optimization problem*

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}\|F(x)\|^2,$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ then

- *Let x be the root of $F(x)$, then x is the minimizer of the optimization problem.*
- *If x satisfies $\nabla f(x) = 0$, and x^** ***is not degenerate***[a]*, then x^* is the root of $F(x)$.*

[a] $x^* \in \mathbb{R}^n$ is called degenerate if the Jacobian $J(x^*)$ is singular.

Proof. (1) If $F(x^*) = 0$, then x^* is a minimizer since $f(x) \geq 0, \forall x$; (2) At local minimizer, we must have $\nabla f = J(x)^T F(x) = 0$, if $J(x^*) \neq 0$, then we must have $F(x) = 0$. □

Remark 7.5.10 (calculating root by minimizing nonlinear least square).

- This lemma enables us to convert nonlinear root problem to nonlinear least square problem.
- After converting to nonlinear least problem, we have two corresponding algorithms (Gauss-Newton algorithm and Levenberg-Marquardt algorithm) to solve the problem.

7.6 Notes on bibliography

Good general references are [1][2][3].

A good reference on Least square problems are [5].

BIBLIOGRAPHY

1. Bertsekas, D. *Nonlinear programming* ISBN: 9781886529007 (Athena Scientific, 2016).
2. Nocedal, J. & Wright, S. *Numerical optimization* (Springer Science & Business Media, 2006).
3. Robinson, D. *Nonlinear optimization I lecture notes* (Johns Hopkins University, 2015).
4. Gentle, J. *Matrix Algebra: Theory, Computations and Applications in Statistics* ISBN: 9783319648675 (Springer International Publishing, 2017).
5. Pighin, F. & Lewis, J. P. *Practical least-squares for computer graphics: Video files associated with this course are available from the citation page* in *ACM SIGGRAPH 2007 courses* (2007), 1–57.

8

CONSTRAINED NONLINEAR OPTIMIZATION

8.1 Quadratic optimization I: equality constraints

8.1.1 Problem formulation

In **equality constrained quadratic optimization**, the optimization problem is given by

$$\min_{x \in \mathbb{R}^n} f(x) = c^T x + \frac{1}{2} x^T H x, \ \textit{subject to } Ax = b,$$

where $c \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$ is symmetric, and $A \in \mathbb{R}^{m \times n}, m \leq n, rank(A) = m, b \in \mathbb{R}^M$. We allow H to be either positive semi-definite or negative definite.

Example 8.1.1.

$$\begin{cases} \max_{x_1, x_2 \in \mathbb{R}} & x_1 x_2 \\ \text{subject to} & x_1 + x_2 = 100 \end{cases}$$

Example 8.1.2.

$$\begin{cases} \max_{x_1, x_2, x_3 \in \mathbb{R}} & x_1^2 + x_1 x_2 + x_2^2 + x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 100 \\ & 2x_2 - x_3 = 10 \end{cases}$$

8.1.2 Optimality condition

8.1.2.1 *General case*

Similar to unconstrained optimization, the first optimality condition can be derived by requiring objective function is non-decreasing on the local tangent space defined by the constraints $Ax = b$.

Theorem 8.1.1 (first order necessary optimality condition, KKT condition). *Consider $x^* \in \mathbb{R}^n$ satisfying feasibility condition $Ax^* = b$. If x^* is a local minimizer of the equality constrained quadratic optimization, then there exists a vector $y^* \in \mathbb{R}^m$ such that*

$$\nabla f(x^*) = A^T y^*.$$

Since $\nabla f(x^*) = Hx^* + c$, we can also equivalently write the two conditions into the so-called ***KKT matrix****:*

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ -y^* \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix}.$$

Proof. We use contradiction to prove. Suppose we cannot find any vector $y^* \in \mathbb{R}^m$ such that $g \triangleq \nabla f(x^*) = A^T y^*$, then we can conclude $\nabla f(x^*) \in \mathcal{R}(A^T)$. Note that we can decompose $g = g_R + g_N$ into a component g_R in $\mathcal{R}(A^T)$ and a **non-zero** component g_N in $\mathcal{N}(A)$. (This decomposition is guaranteed by rank-nullity decomposition theorem [Corollary 5.4.4.1]). Then we can find a nonzero vector $p \in \mathcal{N}(A)$ that $p^T g < 0$,

$$\nabla f(x^* + \alpha p) = f(x^*) + \alpha p^T g + O(\alpha^2) < f(x^*)$$

for sufficiently small α. This contradicts the fact that x^* is a local minimizer. □

We can view A^T as a matrix of column vectors that span the subspace perpendicular to the null space of A, which is the feasible movement for any iterate x. By specifying $\nabla f(x^*) = A^T y^*$, we are requiring the objective function has no tendency of increase or decrease when x move in allowed local region as defined by $A^T x = b$. As an illustration [Figure 8.1.1], we examine the optimality condition for $f(x_1, x_2) = x_1^2 + x_2^2$ under constraint $x_1 + x_2 = 1$. The KKT condition requires the gradient of f should align with the normal of the plane $x_1 + x_2 = 1$ at a local minimal.

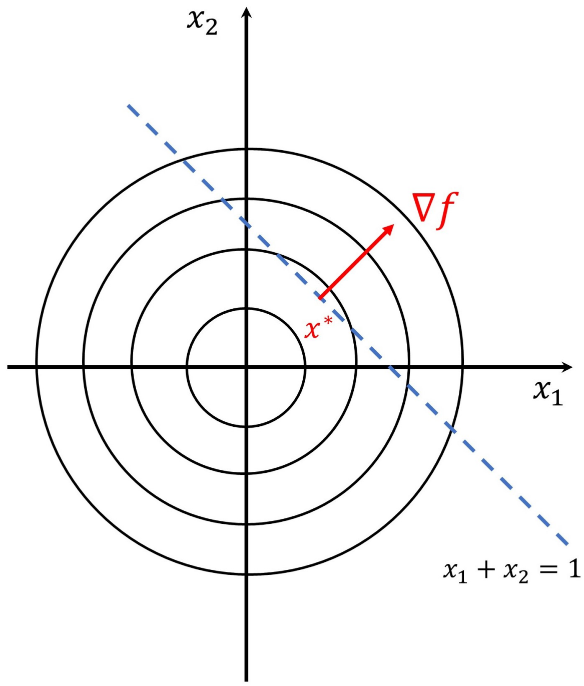


Figure 8.1.1: Demonstration of KKT condition at a local minimal for $f(x_1, x_2) = x_1^2 + x_2^2$ under constraint $x_1 + x_2 = 1$.

Lemma 8.1.1 (condition for nonsingular KKT matrix). *[1, lec 6] If A has full row rank and $Z^T HZ > 0$, where the columns of Z form a basis for the null space of A, then the KKT matrix*

$$K = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}$$

is nonsingular.

Proof. (a) Note that A full row rank is critical. Otherwise, KKT matrix will directly be rank deficient since

$$rank(KKT) = rank(H) + rank(A)$$

. (b) Suppose that

$$K \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

We want to show we must have $u = v = 0$. From above, we have

$$Au = 0, Hu + A^T v = 0$$

. Since Z columns span $\mathcal{N}(A)$, we have $u = Zw$. Multiply $Hu + A^T v = 0$ by u^T, we have

$$u^T Hu + u^T A^T v = w^T Z^T HZw = 0.$$

Since $Z^T HZ > 0$, we have $w = 0, u = Zw = 0$. Then,

$$Hu + A^T v = A^T v = 0 \implies v = 0,$$

since A^T has full column rank.

□

Based on the study of the reduced Hessian, we now have stronger sufficient conditions.

Theorem 8.1.2 (sufficiency of KKT condition in equality constrained quadratic optimization). *For the equality constraint quadratic optimization [Definition 8.3.1], we have the following possibilities:*

- $Z^T HZ > 0$. *There exists an unique minimizer from the unique solution of the KKT solution. Moreover, this unique minimizer must be an isolated/strict/global minimizer. Particularly, if H is positive definite, then* $Z^T HZ > 0$.
- $Z^T HZ \geq 0$, $Z^T HZ$ *is singular, KKT system is consistent but non-unique. Then, the quadratic programming has a unique minimum value but the minimizer is not unique.*
- $Z^T HZ$ *is not positive semidefinite. Then there exists a feasible ray upon which the objective function is unbounded blow.*

Proof. We can turn the constrained optimization problem into a unconstrained quadratic optimization, given by

$$\min_{y \in \mathbb{R}^{n-m}} c^T(x_0 + Zy) + \frac{1}{2}(x_0 + Zy)^T H(x_0 + Zy)$$

where $x_0 + Zy$,Z columns are basis of the $\mathcal{N}(A)$, is the solution to $Ax = b$. Then, we can easily get to the conclusions. □

8.1.2.2 *Positive semi-definitive quadratic programming*

If the objective function $f(x) = c^T x + \frac{1}{2} x^T Hx$ has H being positive semi-definitive, then we have more specific optimality conditions.

Lemma 8.1.2 (local minimum and global minimum in equality constrained quadratic programming). *Let x^* be a local minimizer of the equality constrained quadratic optimization problem.*

- *If H is positive semi-definite, then x^* is also a global minimizer.*
- *If H is positive definite, then the optimization problem only has at most one global minimizer and x^* is the only global minimizer.*

Proof. (1)Let x^* be a local minimum, suppose there is a point $x' \neq x^*$ being a global minimizer such that $f(x') < f(x^*)$. Then

$$f(x') = f(x^* + (x' - x^*)) = f(x^*) + \frac{1}{2}(x' - x^*)^T H(x' - x^*) \geq f(x*),$$

which contradicts that x' is global minimizer. (2) Suppose there are two global minimizer at x_1, x_2 such that $f(x_1) = f(x_2)$.

$$f(x_2) = f(x_1 + (x_2 - x_1)) = f(x_1) + \frac{1}{2}(x_2 - x_1)^T H(x_2 - x_1) > f(x_1),$$

which contradicts $f(x_1) = f(x_2)$. □

Theorem 8.1.3 (first order necessary and sufficient optimal condition for positive semi-definite quadratic programming, KKT condition). *Consider $x^* \in \mathbb{R}^n$ satisfying feasibility condition $Ax^* = b$. Let H be positive semi-definite. Then x^* is a global minimizer of the equality constrained quadratic optimization, if and only if there exists a vector $y^* \in \mathbb{R}^m$ such that*

$$\nabla f(x^*) = A^T y^*.$$

Proof. (1) If x^*, that fact that we have $\nabla f(x^*) = A^T y^*$ is addressed in Theorem 8.1.2. (2) The other direction is addressed in Theorem 8.1.2. □

8.1.3 Solving KKT systems

8.1.3.1 *Factorization approach*

The first way we introduce is to solve the KKT system by empolying symmetric factorization of the KKT matrix and then solving (x, y) sequentially. We have factorization given by

$$P^T K P = L D L^T$$

where P is an appropriately chosen permutation matrix, L is lower triangular with $diag(L) = I$, and D is block diagonal.

Since $K = PLDL^TP^T$, we can use the following procedure to sove the KKT system.

$$\begin{aligned} &\text{solve} \quad Lz = P^T \begin{pmatrix} -c \\ b \end{pmatrix} \\ &\text{solve} \quad D\hat{z} = z \\ &\text{solve} \quad L^T\tilde{z} = \hat{z} \\ &\text{set} \quad \begin{pmatrix} x^* \\ y^* \end{pmatrix} = P\tilde{z} \end{aligned}$$

where we use z to substitute $DL^TP^T \begin{pmatrix} x^* \\ y^* \end{pmatrix}$, $\hat{z}$ to substitute $L^TP^T \begin{pmatrix} x^* \\ y^* \end{pmatrix}$, etc.

8.1.3.2 *Range space approach*

Solving the KKT system involves solving

$$\begin{aligned} Hx - A^Ty &= -c \\ Ax &= b \end{aligned}$$

Since A is full row rank, we can get one solution of x given by $x = (A^TA)^{-1}A^Tb$. Multiply AH^{-1} on both sides of the first equation, we have

$$AH^{-1}Hx - AH^{-1}A^Ty = -AH^{-1}c,$$

which simplifies to

$$b - AH^{-1}A^Ty = -AH^{-1}c,$$

The range approach is quite effective under following conditions:

- B can be easily inverted or B^{-1} is known analytically.
- Small dimensionality problems.

8.1.4 Linear least square with linear constraints

8.1.4.1 *Least norm problem*

Lemma 8.1.3 (least norm with linear constraint). *The constrained minimizing problem*

$$\min \|x\|^2$$
$$\textit{subject to } Cx = d$$

where

- $C \in \mathbb{R}^{p\times n}, d \in \mathbb{R}^p$
- *C has full row rank.*

has solution given by

$$x^* = C^T(CC^T)^{-1}d.$$

Note that $C^T(CC^T)^{-1}$ *is the pseudo-inverse of* C *[Definition 5.14.1).*

Proof. First $Cx^* = CC^T(CC^T)^{-1}d = d$ implies the constraint is satisfied. Second, for any $x \neq x^*, Cx = d$, we have

$$\begin{aligned}\|x\|^2 &= \|x - x^* + x^*\|^2 \\ &= \|x^*\|^2 + \|x - x^*\|^2 + 2(x - x^*)^T x^* \\ &= \|x^*\|^2 + \|x - x^*\|^2 \\ &\geq \|x^*\|^2\end{aligned}$$

where we use the fact that

$$(x - x^*)^T x^* = (x - x^*)^T C^T (CC^T)^{-1} d = (Cx - Cx^*)^T (CC^T)^{-1} d = 0.$$

□

Lemma 8.1.4 (least norm with linear constraint). *The constrained minimizing problem*

$$\min \|Ax - b\|^2$$
$$\textit{subject to } Cx = d$$

where

- $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$
- $C \in \mathbb{R}^{p\times n}, d \in \mathbb{R}^p$
- A, C *has full row rank.*

has

- *the solution pair* (x^*, y^*) *satisfying*
$$Cx^* = d, A^TAx^* + A^Tb = C^Ty^*$$
or equivalently,
$$\begin{pmatrix} A^TA & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} x^* \\ -y^* \end{pmatrix} = \begin{pmatrix} -A^Tb \\ d \end{pmatrix}.$$
- x^* *is given by*
$$x^* = x_0 + (A^TA)^{-1}C^T(C(A^TA)^{-1}C^T)^{-1}(d - Cx_0), x_0 = (A^TA)^{-1}A^Tb.$$

Proof. (1) Theorem 8.1.1 (2) Use block matrix inversion Lemma A.8.6 to get solve the KKT equation.

□

Remark 8.1.1 (interpretation).

- Without the linear constraint, we get the normal equation solution
$$x_0 = (A^TA)^{-1}A^Tb.$$
- The additional linear constraint acts as adjusting the original solution.

8.1.5 Application: Markovitz Portfolio Optimization Model

Definition 8.1.1. *A* ***portfolio vector*** *is a vector* $w \in \mathbb{R}^d$*, with the constraint* $\sum_i w_i = 1$*. We require* $\sum_{i=1}^n w_i = 1$ *can be thought as we split a unit money into investment of different assets.*

Suppose in our universe, there are n stocks. We are further given n estimated return $E(r_i)$, and the covariance matrix $Cov(r_i, r_j) = \Sigma$. For a portfolio characterized by w^Tr, where w is the portfolio vector $w \in \mathbb{R}^n, \sum_{i=1}^n w_i = 1$ and r is the random variable vector $r \in \mathbb{R}^n$ with each component r_i characterized the return rate of asset i, the expected total return and the variance are given as
$$E(w^Tr) = w^TE(r) = \sum_{i=1}^n w_iE(r_i)$$
$$Var(w^Tr) = w^TCov(i,j)w = \sum_{i,j}^n w_iw_j\sigma_{ij}$$

Definition 8.1.2 (mean-variance optimization formulation, minimum variance at fixed return). *Consider n assets with one period return given by $r_1, r_2, ..., r_n$. Denote the return mean by $\mu_1, \mu_2, ..., \mu_n$ and the covariance matrix by $\Sigma, \sigma_{ij} \triangleq \Sigma_{ij}$. Given an arbitrary value μ_0, we want to construct a portfolio,characterized by weight vector $w \in \mathbb{R}^n$ with such return μ_0 and minimum variance. The mean-variance optimization problem is given as*

$$\begin{aligned} \min_w \frac{1}{2} \sum_{i,j}^n w_i w_j \sigma_{ij} \\ s.t. \sum_{i=1}^n w_i \mu_i = \mu_0 \\ \sum_i^n w_i = 1. \end{aligned}$$

Theorem 8.1.4 (sufficient and necessary conditions for efficient portfolio). *Consider n assets with one period return given by $r_1, r_2, ..., r_n$. Denote the return mean by $\mu_1, \mu_2, ..., \mu_n > 0$ and the* ***positive-definite*** *covariance matrix by $\Sigma, \sigma_{ij} \triangleq \Sigma_{ij}$. Consider the mean-variance optimization problem Given an arbitrary value μ_0, we want to construct with such return μ_0 and minimum variance. The mean-variance optimization problem is given as*

$$\begin{aligned} \min_w \frac{1}{2} \sum_{i,j}^n w_i w_j \sigma_{ij} \\ s.t. \sum_{i=1}^n w_i \mu_i = \mu_0 \\ \sum_i^n w_i = 1. \end{aligned}$$

where the portfolio is characterized by weight vector $w \in \mathbb{R}^n$ and μ_0 is preselected number.

It follows that

- *There exists weights* $w_i, i = 1, ..., n$ *and the two Lagrange multipliers* λ_1 *and* λ_2 *satisfying*

$$\sum_{j=1}^{n} \sigma_{ij} w_j - \lambda_1 \mu_i - \lambda_2 = 0, i = 1, ..., n$$

$$\sum_{i=1}^{n} w_i \mu_i = \mu_0$$

$$\sum_{i=1}^{n} w_i = 1$$

- *Because* Σ *is positive definite, there exists unique* $(w, \lambda_1, \lambda_2)$ *to the above linear system; Moreover,* w *is the unique strict global minimizer.*

Proof. See the KKT condition for quadratic optimization Theorem 8.1.2. □

8.2 Quadratic optimization II: inequality constraints

8.2.1 Problem formulation

In inequality constraint quadratic programming, the

$$\min_{x\in\mathbb{R}^n} q(x) = c^T x + \frac{1}{2}x^T H x$$

subject to:

$$A_{\mathcal{E}}x = b_{\mathcal{E}}, A_{\mathcal{I}}x \geq b_{\mathcal{I}},$$

where

- $c \in \mathbb{R}^n, H \in \mathbb{R}^{n\times n}$ and H is symmetric and positive definite, $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$.
- $\mathcal{E} = \{1, 2, ..., m_{\mathcal{E}}\}, \mathcal{I} = \{m_{\mathcal{E}} + 1, ..., m\}$, and $rank(A_{\mathcal{E}}) = m_{\mathcal{E}} \leq n$.

Example 8.2.1.

$$\begin{cases} \max_{x_1,x_2,x_3\in\mathbb{R}} & x_1^2 + x_1x_2 + 2x_2^2 + 3x_3 \\ \text{subject to} & x_1 + x_2 + x_3 = 100 \\ & x_2 + x_3 \geq 5 \end{cases}$$

8.2.2 Optimality conditions

8.2.2.1 *Pure inequality case*

Suppose in our inquality constrained optimization we only have inequality constraints $Ax \geq b$. The idea of first order optimality condition at x^* is that as we move around x^* while satisfying the constraint $Ax \geq b$, the objective function is non-decreasing.

There are two situations. First suppose x^* lies inside the interior of the feasible region defined by $Ax \geq b$, then the optimality condition is the same as a unconstrained optimization problem, i.e., $\nabla f(x^*) = 0$. This is because the constraint is not limiting the movement of x around x^*; that is, the constraint is inactive.

Second, suppose x^* lies on the boundary of the feasible region, then the optimality condition will have the form of $\nabla f(x^*) = A^T y^*, y^* > 0$, which encodes the idea that objective function will not decrease when move around x^* within the feasible region.

In the following, we first studied the Farka's Lemma, which enables us to transform the geometry insight into algebraic conditions. Then we present the KKT condition that accommodates the above two situations.

Lemma 8.2.1 (Farkas's lemma, alternative). *Let $g \in \mathbb{R}^n$ and $A \in \mathbb{R}^{r\times n}$. It follows that*

$$g^T p \geq 0, \forall p \in \{p : Ap \geq 0\}$$

if and only if there exists $\lambda \in \mathbb{R}^r, \lambda \geq 0$ such that

$$g = A^T \lambda.$$

Proof. (1) forward. If $g = A^T\lambda$, then $g^T p = \lambda A p \geq 0$; (2) converse. If $g \notin \mathcal{X} = \{A^T\lambda, \lambda \geq 0\}$, then g is the point lying outside the cone $\mathcal{X}$. Based on Theorem 10.2.3, there exists p such that $Ap \geq 0$(all the basis vectors of the cone lying on one halfspace of a hyperplane passing origin and having norm vector p) and $g^T p < 0$(the element g lying on the other halfspace). This contradicts that for all p such that $Ap \geq 0$, $g^T p \geq 0$. □

Theorem 8.2.1 (KKT optimality condition for pure inequality constrained quadratic programming). *If x^* is a minimizer of quadratic programming [subsection 8.2.1], then there exists $y \in \mathbb{R}^m$ such that*

$$\begin{aligned} a_i^T x^* &\geq b_i, \forall i \in \mathcal{I} \\ y_i^* &= 0, \forall i \in \mathcal{I}/\mathcal{A}(x^*) \\ y_i^* &\geq 0, \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \\ Hx^* + c = A^T y^* &= \sum_{i\leq i\leq m} y_i^* a_i \end{aligned}$$

where we use a_i^T to denote the ith row of A, $\mathcal{A}(x^*)$ denotes the active/binding index set at x^*, defined as

$$\mathcal{A}(x^*) = \{i : a_i^T x^* = b_i, \forall i \in \mathcal{I}\}$$

Proof. We consider a small movement step δx from x^*. To ensure $x^* + \delta x$ to remain in the feasible region, we require $A\delta x \geq 0$. Because x^* is a local minimizer, the gradient $g = Hx^* + c$ must satisfy $g^T\delta \geq 0$ (we can prove this via contradiction argument, similar to Theorem 8.1.2). Using Farkas' Lemma [Lemma 8.2.1], we must have $g = A^T y^*, y^* \geq 0$. For constraints that are inactive, we can set their associated $y*$ to zero. □

As an illustration [Figure 8.2.1], we examine the optimality condition for $f(x_1, x_2) = x_1^2 + x_2^2$ under constraint $x_1 + x_2 \geq 1$. The KKT condition for a local minimal requires the decreasing of f cannot be achieved when x moves inside the feasible region.

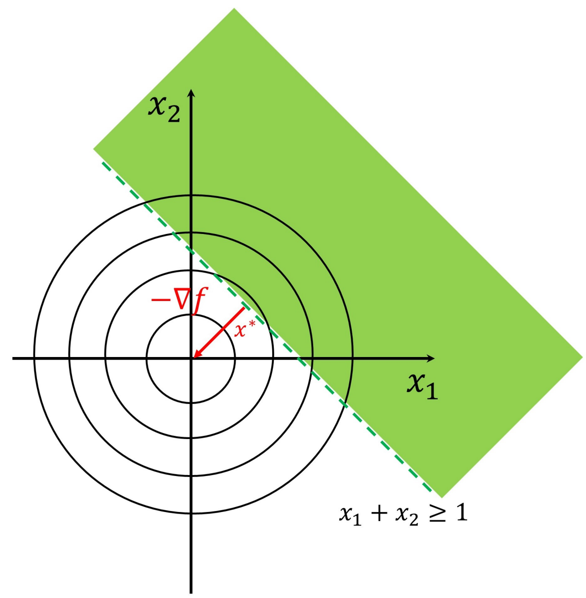


Figure 8.2.1: Demonstration of KKT condition at a local minimal for $f(x_1, x_2) = x_1^2 + x_2^2$ under constraint $x_1 + x_2 \geq 1$.

8.2.2.2 *General constrained optimization*

We can combine KKT conditions of equality constraint case and pure inequality constraint case and arrive at the following KKT conditions for general constrained quadratic optimization.

Theorem 8.2.2 (KKT optimality condition general constrained quadratic programming). *If x^* is a minimizer of quadratic programming [subsection 8.2.1], then there exists $y \in \mathbb{R}^m$ [a] such that*

$$\begin{aligned} a_i^T x^* &= b_i, \forall i \in \mathcal{E} \\ a_i^T x^* &\geq b_i, \forall i \in \mathcal{I} \\ y_i^* &= 0, \forall i \in \mathcal{I}/\mathcal{A}(x^*) \\ y_i^* &\geq 0, \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \\ Hx^* + c = A^T y^* &= \sum_{i \leq i \leq m} y_i^* a_i \end{aligned}$$

where $\mathcal{A}(x^*)$ denotes the active set at x^*, defined as

$$\mathcal{A}(x^*) = \mathcal{E} \cup \{i : a_i^T x^* = b_i, \forall i \in \mathcal{I}\}$$

a for y_i associated with equality constraints, y_i will not subject to non-negative constraint.

8.2.2.3 *Positive semi-definitive quadratic programming*

Lemma 8.2.2 (local minimum and global minimum in inequality constrained quadratic programming). *Let x^* be a local minimizer of the equality constrained quadratic optimization problem.*

- *If H is positive semi-definite, then x^* is also a global minimizer.*
- *If H is positive definite, then the optimization problem only has at most one global minimizer and x^* is the only global minimizer.*

Proof. Because the feasible regions are a convex set, we can use same proof technique in Lemma 8.1.2. □

Theorem 8.2.3 (KKT necessary and sufficient optimality condition for positive semi-definite quadratic programming). *If x^* is a minimizer of quadratic programming [subsection 8.2.1), then there exists $y \in \mathbb{R}^m$ such that*

$$\begin{aligned} a_i^T x^* &= b_i, \forall i \in \mathcal{E} \\ a_i^T x^* &\geq b_i, \forall i \in \mathcal{I} \\ y_i^* &= 0, \forall i \in \mathcal{I}/\mathcal{A}(x^*) \\ y_i^* &\geq 0, \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \\ Hx^* + c = A^T y^* &= \sum_{i \leq i \leq m} y_i^* a_i \end{aligned}$$

where $\mathcal{A}(x^*)$ denotes the active set at x^*, defined as

$$\mathcal{A}(x^*) = \mathcal{E} \cup \{i : a_i^T x^* = b_i, \forall i \in \mathcal{I}\}$$

Moreover, if H is semi-positive definite, then above condition for x^* is sufficient for being global minimizer.

Proof. (1) Forward direction is from KKT condition [Theorem 8.2.2]. (2) Consider another feasible point $x' = x^* + p$ where $p \neq 0, p^T a_i = 0, \forall i \in \mathcal{E}, p^T a_i \geq 0, \forall i \in \mathcal{A} \cap \mathcal{I}$. Then

$$\begin{aligned} & q(x^* + p) - q(x^*) \\ =& \frac{1}{2} p^T H p + p^T c + p^T H x^* \\ =& \frac{1}{2} p^T H p + p^T (A^T y^*) \\ =& \frac{1}{2} p^T H p + \sum_{i \in \mathcal{I}} p^T a_i y_i^* \geq 0 \end{aligned}$$

where we use the fact that $y_i^* = 0, \forall i \in \mathcal{I}/\mathcal{A}$ (inactive constraint), and $p^T a_i \geq 0, \forall i \in \mathcal{A} \cap \mathcal{I}$. Because p is arbitrarily chosen, therefore x^* is a local minimum. □

8.2.3 Primal active-set method

Definition 8.2.1 (working set). *A working set $\mathcal{W}_k$ is a set of index such that*

$$\mathcal{E} \subseteq \mathcal{W}_k \subseteq \mathcal{A}(x_k)$$

Definition 8.2.2 (subspace minimizer). *Given the working set $\mathcal{W}_k$ associated with x_k, we say that the solution x_k^* to*

$$\min_{x \in \mathbb{R}^n} q(x) = c^T x + \frac{1}{2} x^T H x, \text{ subject to } A_{\mathcal{W}_k} x = b_{\mathcal{W}_k}$$

is a subspace minimizer associated with $\mathcal{W}_k$, where $A_{\mathcal{W}_k}$ denotes the rows from A correspond to $\mathcal{W}_k$.

Lemma 8.2.3 (step to subspace minimizer). *Given current iterate x_k, the step p_k to the subspace minimizer such that $x_k^* = x_k + p_k$ is also the minimizer of the following minimizer problem*

$$\min_{p \in \mathbb{R}^n} \frac{1}{2} p^T H p + g_k^T p, \text{ subject to } A_{\mathcal{W}_k} p = 0,$$

where $g_k^T = \nabla q(x_k)$.

Proof.

$$\begin{aligned} q(p+x_k) &= c^T(p+x_k) + \frac{1}{2}(p+x_k)^T H(p+x_k) \\ &= c^T p + c^T x + \frac{1}{2}p^T H p + p^T H x_k = p^T g_k + \frac{1}{2}p^T H p \end{aligned}$$

where $g_k = Hx_k + c$. □

Lemma 8.2.4 (subspace minimizer step as descend step). *Let p_k be the subspace minimizer associated with constraint matrix $A_{\mathcal{W}_k}$. Assume that $p_k \neq 0$ and $H > 0$, then*

- $g_k^T p_k = -p_k^T H p_k < 0.$
- $q(x_k + \alpha p_k) < q(x_k)$ *for all* $0 < \alpha < 2.$

Proof. (1) The KKT condition for p_k to be a minimizer of

$$\min_{p\in\mathbb{R}^n} \frac{1}{2}p^T H p + g_k^T p, \text{ subject to } A_{\mathcal{W}_k} p = 0,$$

is given by

$$Hp_k - A_{\mathcal{W}_k}^T y = -g_k, A_{\mathcal{W}_k} p_k = 0.$$

Then,

$$p_k^T g_k = -p_k^T H p_k + p_k^T A_{\mathcal{W}_k}^T y = -p_k^T H p_k < 0.$$

(2)

$$q(x_k+\alpha p_k) - q(x_k) = \frac{1}{2}\alpha^2 p_k^T H p_k + \alpha p^T g_k = \frac{1}{2}\alpha^2 p_k^T H p_k - \alpha p_k^T H p_k = \frac{1}{2}\alpha(\alpha-2)p_k^T H p_k,$$

when $2 > \alpha > 0$, we have $q(x_k + \alpha p_k) < q(x_k)$. □

Remark 8.2.1 (overshooting issue). Note that when we take $1 < \alpha < 2$, we are moving more than p_k, but we can still achieve decrease in the q due to the quadratic nature of q.

Lemma 8.2.5 (decreasing and blocking constraints in the search and descent direction). *Let p_k be the subspace minimizer associated with constraint matrix $A_{\mathcal{W}_k}$. Assume that $p_k \neq 0$ and $H > 0$.*

The set of ***decreasing and blocking constraints*** *$\mathcal{D}_k$ at the point x_k is given by*

$$\mathcal{D}_k \triangleq \{j \in [1:m] : j \notin \mathcal{W}_k, a_j^T p_k < 0\},$$

where a_j^T is the jth row of the constraint matrix.

The maximum step size α before hitting a blocking constraint i is given by:

$$\alpha = \frac{b_i - a_i^T p_k}{a_i^T p_k}.$$

Note that constraint $i \in \mathcal{W}_k$ will not be a blocking constraint since $a_i^T p_i = 0$.

Proof. Note that

$$a_i^T(x_k + \alpha p_k) = b_i \implies \alpha = \frac{b_i - a_i^T p_k}{a_i^T p_k}.$$

□

Remark 8.2.2 (geometry of decreasing constraints)**.** When we move along direction p_k, we might encounter hyperplanes(i.e. other constraints). The set of hyperplanes encountered are $\mathcal{D}_k$.

Moreover, if we do not encounter any hyperplanes, then we can take the minimizer step p_k.

Algorithm 12: Primal active-set method for strictly convex quadratic programming

Input: Initial feasible x_0 with associated working set $\mathcal{W}_0$ such that $\mathcal{E} \subseteq \mathcal{W}_0 \subseteq \mathcal{A}(x_0)$

Set $k = 0$

repeat

  Compute gradient $g_k = \nabla q(x_k) = Hx_k + c$.

  Compute y_k as $[y_k]_{i \notin \mathcal{W}_k} = 0$ and $[y_k]_{\mathcal{W}_k} = \hat{y}_k$, where $(p_k, \hat{y}_k)$ as the solution to

$$\min_p \frac{1}{2} p^T H p + g_k^T p, \text{ subject to } A_{\mathcal{W}_k} p = 0.$$

  if $p_k = 0$ **then**

    if $[y_k]_i \geq 0$ *for all* $i \in \mathcal{W}_k \cap \mathcal{I}$ **then**

      return x_k as the minimizer since KKT condition is satisfied

    end

    else

      Set $s = \arg\min_{i \in \mathcal{W}_k \cap I} [y_k]_i, x_{k+1} = x_k, \mathcal{W}_{k+1} = \mathcal{W}_k / \{s\}$,

    end

  end

  else

    Set $x_{k+1} = x_k + \alpha_k p_k$ and α_k is the minimum step length such that

$$\alpha_k = \min(1, \min_{i \in \mathcal{D}_k} \frac{b_i - a_i^T x_k}{a_i^T p_k}),$$

    where $\mathcal{D}_k = \{i \in [1:m] : i \notin \mathcal{W}_k, a_i^T p_k < 0\}$.

    if $\alpha_k < 1$ **then**

      set $\mathcal{W}_{k+1} = \mathcal{W}_k \cup \{t\}$ for some $t \in \mathcal{D}_k$ satisfying $\frac{b_t - a_t^T x_k}{a_t^T p_k} = \alpha_k$

    end

    else

      set $\mathcal{W}_{k+1} = \mathcal{W}_k$.

    end

  end

  Set $k = k + 1$.

until *termination condition satisfied*;

Output: approximate minimizer x_k

Remark 8.2.3 (interpretation).

- We set $[y_k]_{i\notin\mathcal{W}_k} = 0$ means that we assume inequality constraints not in the working set $\mathcal{W}_k$ will be set as inactive.
- If $p_k = 0$ and $[y_k]_i \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$, then the KKT condition is satisfied; If $p_k = 0$, but not $[y_k]_i \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$, then that mean by stepping off some active constraint, we can get objective function to decrease(see the following lemma).
- If $p_k > 0$, then we can show that p_k is a descent direction [Lemma 8.2.4], so we move along p_k until a constraint is hit.

Lemma 8.2.6 (feasible direction when stepping off constraints). *[2, p. 470] Let p_k be the subspace minimizer associated with constraint matrix $A_{\mathcal{W}_k}$. Assume*

- $p_k = 0$
- $\{a_i\}_{i\in\mathcal{W}_k}$ *is a linearly independent set*
- *there exists some $j \in \mathcal{W}_k \cap \mathcal{I}$ such that $[y_k]_j < 0$.*

Then the direction p_{k+1} computed as the solution to

$$\min_{p\in\mathbb{R}^n} \frac{1}{2}p^T Hp + g_k^T p, \text{ subject to } a_i^T p = 0, \forall i \in \mathcal{W}_k/\{j\}$$

satisfies $a_j^T p_k \geq 0$, i.e., it is a feasible direction for constraint j.

Moreover, if $H > 0$, then

$$a_j^T p_k > 0,$$

satisfies s

Proof. Note that $p_k = 0$, we have

$$Hp_k = 0 = -g + \sum_{i\in\mathcal{W}_k} y_i a_i.$$

For p_{k+1}, we have

$$Hp_{k+1} = 0 = -g + \sum_{i\in\mathcal{W}_k-\{j\}} y_i' a_i.$$

Subtract, we get

$$-Hp_{k+1} = \sum_{i\in\mathcal{W}_k-\{j\}} (y_i - y_i')a_i + a_j^T y_j.$$

Multiply both sides by p_{k+1}, we get

$$0 > -p_{k+1}Hp_{k+1} = p_{k+1}^T a_j y_j \implies p_{k+1}^T a_j > 0.$$

□

Remark 8.2.4. This lemma says if we encounter the situation $p_k = 0$ but there exists negative multiplier in the active constraint. Then we should remove the constraint and solve the subspace minimizer problem again. The new subspace minimizer will be descent directionLemma 8.2.4 and step off the constraint to a feasible region.

8.2.4 Gradient projection method

Definition 8.2.3 (bound-constrained quadratic optimization). *The bound-constrained quadratic optimization problem is given by*

$$\min_{x \in \mathbb{R}^n} q(x) = c^T x + \frac{1}{2} x^T H x, \text{ subject to } l \leq x \leq u$$

where

- $c \in \mathbb{R}^n$ *and* $H \in \mathbb{R}^{n \times n}$ *is symmetric.*
- H *may be indefinite, i.e.,* q *may be nonconvex.*
- $l \in \mathbb{R}^n$*(may contain* $-\infty$*) and* $u \in \mathbb{R}^n$*(may contain* ∞*). Assume* $l < u$.

Definition 8.2.4 (projection operator). *The projection operator* $P : \mathbb{R}^n \to \mathbb{R}^n$ *associated with the constraints* $l \leq x \leq u$ *is defined component-wise as*

$$[P(x; l, u)]_i = \begin{cases} l_i, & if\ x_i < l_i \\ x_i, & if\ l_i \leq x_i \leq u_i \\ u_i, & if\ x_i > u_i \end{cases}$$

Remark 8.2.5. **the range of the projection operator is the feasible set**

Definition 8.2.5 (projected gradient path). *The projected gradient path is defined by the piece-wise linear path* $\{x(t) : t \geq 0\}$ *such that*

$$x(t) = P(x - t\nabla q(x); l, u), \forall t \geq 0.$$

Definition 8.2.6 (Cauchy step). *The Cauchy step* x^c *is defined as*

$$x^c = x(t^c),$$

where t^c is the solution of [a]

$$\min_{t\geq 0} q(x(t)).$$

[a] The detailed calculation procedure for the Cauchy step can be found at [2, p. 486].

Algorithm 13: First order gradient projection algorithm

Input: Initial $x_{in} \in \mathbb{R}^n$

Set $x_0 = P(x_{in}; l, u)$ to make x_0 feasible. Set $k = 0$
repeat
  if *x_k is first order KKT point* **then**
    return x_k as the minimizer
  end
  With $x = x_k$, compute the Cauchy point x^c.
  Set $x_{k+1} = x^c$.
  Set $k = k + 1$.
until *termination condition satisfied*;

Output: approximate minimizer x_k

8.2.5 Dual convex quadratic programming

Lemma 8.2.7 (dual form inequality constraint convex quadratic optimization). *[3, p. 437] The primal problem of*

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^T Hx + c^T x, \text{ subject to } Ax \geq b$$

with $H > 0$, $A \in \mathbb{R}^{m\times n}$, and $rank(A) = n > m$ has the dual form given by

$$\max_{y\in\mathbb{R}^m} -\frac{1}{2}(A^T y - c)^T H^{-1}(A^T y - c) + b^T y, \text{ subject to } y \geq 0$$

Proof. The Lagrange function is

$$L(x, y) = \frac{1}{2}x^T Hx + c^T x - y^T(Ax - b), y \geq 0.$$

To minimize over x, we set

$$\nabla_x L(x, y) = 0 \implies Hx + c - A^T y = 0.$$

Then, the dual function

$$d(y) = \min_x L(x, y) = -(A^T y - c)^T H^{-1}(A^T y - c) + b^T y, y \geq 0.$$

□

Remark 8.2.6 (convert to easier bound-constrained optimization). Note that using the dual formulation we can convert the convex quadratic optimization to a simplier bound-constrained convex quadratic optimization [Definition 8.2.3].

Lemma 8.2.8 (dual form equality constraint convex quadratic optimization). *The primal problem of*

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^T Hx + c^T x, \text{ subject to } Ax = b$$

with $H > 0$, $A \in \mathbb{R}^{m\times n}$, *and* $\mathrm{rank}(A) = m \leq n$ *has the dual form given by*

$$\max_{y\in\mathbb{R}^m} -\frac{1}{2}(A^T y - c)^T H^{-1}(A^T y - c) + b^T y$$

Proof. Similar to inequality constraint case. □

Remark 8.2.7 (convert to easier unconstrained problem). Note that using the dual formulation we can convert the convex quadratic optimization to a simplier unconstrained optimization.

8.3 General equality constrained optimization

Notations

- Jacobian $J(x) = \nabla c \in \mathbb{R}^{m \times n}$, each row is ∇c_i^T

8.3.1 Feasible path and optimality

Definition 8.3.1 (Equality-constraint optimization). *A equality constrained linear programming is given as:*

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$\textit{subject to } c(x) = 0$$

where $f : \mathbb{R}^n \to \mathbb{R}, c : \mathbb{R}^m \to \mathbb{R}^n$ *where* $c = [c_1, ..., c_m]^T, c_i : \mathbb{R}^n \to R$.

Definition 8.3.2 (feasible path, tangent vector). *A feasible path is a curve for constraints* $c(x) = 0$*, represented by a twice continuously differentiable function* $x(t)$*, that emanates from a feasible point* x_0 *such that*

$$x(0) = x_0, c(x(t)) = 0$$

for all $0 \leq t < \sigma, \sigma > 0$*, and such that* $dx/dt|_{t=0} \neq 0$ *The tangent vector of the feasible path is given as*

$$p = dx(t)/dt|_{t=0}.$$

Definition 8.3.3 (tangent cone). *Given constraints* $c(x) = 0$*, the set*

$$\mathcal{T}(x) = \{p : p \textit{ is a nonzero vector tangent to a feasible path emanating from } x\} \cup \{0\}\}$$

is called the tangent cone of c *at the point* x.

Remark 8.3.1 (tangent cone is a cone). The tangent cone of constraints $c(x) = 0$ at x is a cone; that is, given $p \in \mathcal{T}(x), \alpha p \in \mathcal{T}(x), \alpha > 0$. Note that p is vector tangent to some feasible path emanating from x, then αp will still be a vector tangent to the original path.

Theorem 8.3.1 (first order necessary condition, geometric form). *[1, lec 4]If x^* is a local minimizer, then $c(x^*) = 0$ and $\nabla f(x^*)^T p \geq 0, \forall p \in \mathcal{T}(x^*)$.*

Proof. Suppose for some p, we have $\nabla f(x^*)^T p < 0$, then in direction p, let $\gamma(t)$ be the feasible curve emanating from x^* with tangent vector p, we have

$$f(\gamma(\alpha)) = f(x^*) + \alpha f(x^*)^T p + O(\alpha^2) < f(x^*)$$

as $\alpha \to 0, \alpha > 0$, which contradicts that fact that $f(x^*)$ is local minimum. □

8.3.2 Constraint qualification and Lagrange theory

Lemma 8.3.1 (algebraic characterization of tangent cone).

$$\mathcal{T}(x) \subseteq \mathcal{N}(J(x)),$$

that is, if $p \in \mathcal{T}(x)$, then $(\nabla c_i)^T p = 0, \forall i = 1, ..., m$, or equivalently, $p \in \mathcal{N}(J(x))$.

Proof. Since $c(x(\alpha)) = 0$, for $\alpha \in [0, \sigma)$ along a feasible path $x(\alpha)$ emanating from x, we know that

$$0 = \frac{d}{d\alpha} c_i(x(\alpha))|_{\alpha=0} = \nabla c_i(x)^T p$$

where

$$p = \frac{d}{d\alpha} x(\alpha)|_{\alpha=0} = \lim_{\alpha \to 0^+} \frac{x(\alpha) - x(0)}{\alpha}.$$

Therefore, we know that p satisfy $(\nabla c_i)^T p = 0, i = 1, ..., m$, which implies

$$\sum_{i=1}^{m} (\nabla c_i)^T p = 0 \Leftrightarrow Jp = 0$$

Note that $Jp = \sum_{i=1}^{m} (\nabla c_i)^T p = 0$ can not generally implies $(\nabla c_i)^T p = 0, \forall i = 1, ..., m$ unless J has linearly independent rows. □

Example 8.3.1. [4] Consider constraints $c_1(x) = x_1^3 - x_2 = 0, c_2(x) = x_2 = 0$. At a feasible point $x^* = 0$, we have

$$J = \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix},$$

and

$$\mathcal{N}(J(x^*)) = \{(\gamma, 0)^T, \gamma \in \mathbb{R}\}.$$

However, x^* is the only feasible point so that no feasible paths exist.

Therefore, we have

$$\{0\} = \mathcal{T}(x^*) \subset \mathcal{N}(J(x^*)) = \{(\gamma, 0)^T, \gamma \in \mathbb{R}\}.$$

Definition 8.3.4 (constraint qualification). *We say that the constraint qualification of equality constraint $c(x) = 0$ holds at a feasible point x if every nonzero vector p satisfying $J(x)p = 0$ implies $p \in \mathcal{T}(x)$.*

Remark 8.3.2 (purpose of constraint qualifications)**.**

- constraint qualifications is simple the assumptions on constraint such that later KKT condition relies on
- There are different types of constraint qualifications, such as linear independence constraint qualification, Managasarian-Fromovitz constraint qualification.

Lemma 8.3.2. *If constraint qualifications of equality constraint $c(x) = 0$ holds, then*

$$\mathcal{T}(x) = \mathcal{N}(J(x))$$

Proof. directly form above lemma and the definition and constraint qualification. □

Lemma 8.3.3 (linear constraints always satisfy constraint qualification). *The constraint qualification of equality constraint $c(x) = 0$ holds at x if $c(x) = Ax - b = 0$(no matter A has full row rank or not).*

Remark 8.3.3. Assume $Ax = b$ has infinitely many solution, which form a subspace. Consider a feasible point x_0, then from the definition, $\mathcal{T}(x^*)$ is just $\mathcal{N}(A)$ since $A(x + \alpha p) = b, \forall \alpha \geq 0, p \in \mathcal{N}(A)$.

Lemma 8.3.4 (sufficient condition for constraint qualifications, nonlinear constraint case). *[2, p. 324] The constraint qualification of equality constraint $c(x) = 0$ holds at x if $J(x)$ has full row rank (i.e. each row, ∇c_i, are linearly independent of each other).*

Lemma 8.3.5 (existence of Lagrange multipliers for equality constraints). *[1, lec 4] Assume that f and c are differentiable at a feasible point x^*. Then*

$$\nabla f(x^*)^T p \geq 0, \forall p \in \mathcal{N}(J(x^*))$$

if and only if $\nabla f(x^*) \in \mathcal{R}(J(x^*))$; that is there exist some λ such that

$$\nabla f(x^*) = J(x^*)^T \lambda$$

Proof. (1) forward. If $\nabla f(x^*) \in \mathcal{R}(J(x^*))$, then $\nabla f(x^*)^T p = 0$, since $\mathcal{R}(J(x^*)^T) \perp \mathcal{N}(J(x^*))$; (2) converse. Let $\nabla f(x^*) = g = g_N + g_R$ where $g_R \in \mathcal{R}(J(x^*)), g_N \in \mathcal{N}(J^T)$. (This decomposition is guaranteed by rank-nullity decomposition theorem). Then suppose $g \notin \mathcal{R}(J)$, which implies $g_N \neq 0$. Then $g^T p = -g_N^T p < 0$ if $p = g_N \in \mathcal{N}(J^T)$. That is, there exist some p such that $\nabla f(x^*)^T p < 0$. □

Remark 8.3.4. This lemma enables us to formulate our first-order optimality condition more concisely.

Theorem 8.3.2 (first order necessary condition, KKT condition, equality constraints). *Assume that the constraint qualification holds. It follows that if x^* is a local minimizer, then*

$$c(x^*) = 0 \text{ and } g(x^*) = J(x^*)^T \lambda,$$

for some vector λ^*, known as Lagrange multiplier. The pair (x^*, λ^*) is also known as KKT points.

Proof. When constraint qualification holds and x^* is a local minimizer, we know that

$$c(x^*) = 0, \nabla f(x^*)^T p \geq 0, \forall p \in \mathcal{N}(J(x^*)).$$

Then follow the above lemma, we have

$$c(x^*) = 0 \text{ and } g(x^*) = J(x^*)^T \lambda.$$

□

Example 8.3.2. Consider an equality constrained quadratic optimization is given by

$$\min_{x \in \mathbb{R}^n} f(x) = c^T x + \frac{1}{2} x^T H x, \text{ subject to } Ax = b,$$

where $c \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$ is symmetric, and $A \in \mathbb{R}^{m \times n}, m \leq n, rank(A) = m, b \in \mathbb{R}^M$.

The first-order optimality conditions for a minimizer x^* of the equality constrained quadratic optimization problem is that there exists a vector $y^* \in \mathbb{R}^m$ such that

$$Ax^* = b(feasible) \text{ and } Hx^* + c = A^T y^*.$$

Note 8.3.1 (interpretation of Lagrange multipliers, equality constraint). Assume linear independence qualification holds. Suppose that (x^*, λ^*) is a KKT point such that

$$g(x^*) = J(x^*)^T \lambda^* \Leftrightarrow g \perp \mathcal{N}(J).$$

We have the following observations:

- If $\lambda_i^* = 0$, then the constraint i is redundant; that is solution will not change if we ignore this constraint.
- Consider an arbitrary direction $p \in \mathbb{R}^n$:
  - If $J(x)p = 0$, then $p \in \mathcal{N}(J(x^*))$, $g(x^*)^T p = \lambda^T J p = 0$; that is, p is in the feasible direction, however, since x^* is the local minimizer, moving in p will not change the value of objective function.
  - If p is such that $Jp = e_i$(e_i is the ith standard basis), then moving in direction p will only step off the constraint i but maintain within tangent space of other constraints. To see this, we have $c(x + \alpha p) \approx c(x) + e_i$. Moreover, if $y_i > 0$, then $g^T p = y^T J p = y_i$, then moving in direction p will increase the value of objective function; vice versa.
  - If p is such that $Jp \geq 0$, then

    $$g^T p = [y^*]^T J p,$$

    whose value depends on specific signs of components in y^*.
- Note that when we want to examine the effect of a specific constraint on the optimal objective function value, we need to find a direction only step off this specific constraint.
- If $J(x^*)$ has full row rank, then y^* is **unique** since $g = J^T y$.

Remark 8.3.5 (how to calculate (x^*, λ^*)). Let us assume $x \in \mathbb{R}^n$ and $\lambda^* \in \mathbb{R}^m$(i.e. there are m constraints). The feasibility condition $c(x^*)$ provides m equations, the condition $g(x^*) = J(x^*)^T \lambda$ provides n equations. In principle, we can solve (x^*, λ^*) by solving roots to nonlinear equations.

8.3.3 Second order condition

Theorem 8.3.3 (second order necessary condition, geometric form). *[1, lec 4] Assume that the first-order constraint qualification holds. Suppose x^* is a constrained minimizer and consider a feasible path $x(\alpha)$ such that*

$$x(0) = x^*, 0 \neq p \triangleq \frac{d}{d\alpha}x(\alpha)|_{\alpha=0}, v \triangleq \frac{d^2}{d\alpha^2}x(\alpha)|_{\alpha=0}.$$

Then

$$\frac{d^2}{d\alpha^2}f(x(\alpha)) \geq 0,$$

for all such feasible paths.

Proof. Use Taylor expansion along the path $x(\alpha)$, we have

$$\begin{aligned} f(x(\alpha)) &= f(x(0)) + \alpha\frac{d}{d\alpha}f(x(\alpha))|_{\alpha=0} + \frac{\alpha^2}{2}\frac{d^2}{d\alpha^2}f(x(\alpha))|_{\alpha=0} + O(\alpha^3) \\ &= f(x^*) + \alpha g(x^*)^T p + \frac{\alpha^2}{2}\frac{d^2}{d\alpha^2}f(x(\alpha))|_{\alpha=0} + O(\alpha^3) \end{aligned}$$

When the constraint qualification holds, we have

$$g(x^*) = J^T y*, Jp = 0 \implies \alpha g(x^*)^T p = 0.$$

Therefore,

$$f(x(\alpha)) = f(x^*) + \frac{\alpha^2}{2}\frac{d^2}{d\alpha^2}f(x(\alpha))|_{\alpha=0} + O(\alpha^3).$$

Use contradiction argument, we can show that

$$\frac{d^2}{d\alpha^2}f(x(\alpha))|_{\alpha=0} \geq 0.$$

□

Theorem 8.3.4 (second order necessary conditions, algebraic form). *Assume that the constraint qualification holds at x^*. It follows that if x^* is a local minimizer, then*

- *(feasible) $c(x^*) = 0$.*
- *There exists a Lagrange multiplier $y^* \in \mathbb{R}^m$ such that $\nabla f = J(x^*)^T y^*$.*
- *For y^* in (2), we have*

$$p^T[\sum_{i=1}^m -y_i^* \nabla^2 c_i(x^*) + H(x^*)]p \geq 0$$

holds for all $p \in \mathcal{N}(J(x^*))$.

Proof. Use that fact that

$$\frac{d^2}{d\alpha^2} f(x(\alpha)) = \frac{d}{d\alpha}[g(x(\alpha)^T x'(\alpha)] = g(x(\alpha))^T x''(\alpha) + x'(\alpha)^T H(x(\alpha))x'(\alpha)$$

When the constraint qualification holds, we have

$$g(x^*) = J^T y*.$$

Therefore, use Theorem 8.3.3, we have

$$0 \leq \frac{d^2}{d\alpha^2} f(x(\alpha))|_{\alpha=0} = [y^*]^T J(x^*)v + p^T H(x^*)p = \sum_{i=1}^m y_i^* (\nabla c_i(x^*))^T v + p^T H(x^*)p.$$

For a feasible path satisfying $c_i(x(\alpha)) = 0, \forall i$, we have

$$0 = \frac{d}{d\alpha} c_i(x(\alpha)) = \frac{d}{d\alpha}[\nabla c_i(x(\alpha))^T x'(\alpha)] = \nabla c_i(x(\alpha))^T x''(\alpha) + x'(\alpha)^T \nabla^2 c_i(x(\alpha))x'(\alpha).$$

Simplify, we have

$$0 = \nabla c_i(x^*)^T v + p^T \nabla^2 c_i(x^*)p.$$

Then,

$$0 \leq \frac{d^2}{d\alpha^2} f(x(\alpha))|_{\alpha=0} = \sum_{i=1}^m y_i^* (\nabla c_i(x^*))^T v + p^T H(x^*)p$$

will reduce to

$$p^T [\sum_{i=1}^m -y_i^* \nabla^2 c_i(x^*) + H(x^*)]p \geq 0.$$

□

Remark 8.3.6 (interpretation, analog to unconstrained optimization using reduced Hessian)**.**

- The Hessian condition in (3) says that the Hessian of the Lagrange is positive definite for vectors p of linear feasible direction set(i.e. $p \in \mathcal{N}(J(x^*))$).
- If we define $Z(x^*)$ to be the matrix whose columns form a basis of $cN(J(x^*))$, then condition (3) is equivalent to
$$Z^T H Z > 0.$$
- The matrix $Z^T H Z$ is called **reduced Hessian**, and plays the role analogous to $H(x^*)$ in unconstrained optimization.

Theorem 8.3.5 (second order sufficient optimality condition, strict local minimizer). *[1, lec 4][3, p. 364] The vector $x^* \in \mathbb{R}^n$ is a strict local minimizer of the optimization problem [Definition 8.3.1] if*

- *(feasible) $c(x^*) = 0$.*
- *There exists a Lagrange multiplier $y^* \in \mathbb{R}^m$ such that $\nabla f = J(x^*)\lambda^*$.*
- *For y^* in (2), we have*

$$p^T[\sum_{i=1}^{m} -y^* \nabla^2 c_i(x^*) + H(x^*)]p > 0$$

  holds for all $p \neq 0$ and $J(x^*)p = 0$.

In fact, if the above conditions holds, it can be showed that there exists scalars $\gamma > 0$ and $\epsilon > 0$ such that

$$f(x) \geq f(x^*) + \frac{\gamma}{2}\|x - x^*\|, \forall x \text{ such that } h(x) = 0, \|x - x^*\| < \epsilon.$$

Proof. **Note that we do not require constraint qualification.** all feasible directions $\mathcal{T} \subseteq \mathcal{N}(J(x^*))$. We can use similar techniques used in proving necessary conditions. □

8.4 General inequality constrained optimization

Notations

- Jacobian $J(x) = \nabla c \in \mathbb{R}^{n\times m}$, each column is ∇c_i
- Active set $\mathcal{A}(x) = \{i : c_i(x) = 0\}$

8.4.1 Feasible path and optimality

Definition 8.4.1 (inequality-constraint optimization). *A equality constrained linear programming is given as:*

$$\min_{x\in\mathbb{R}^n} f(x)$$
$$\text{subject to } c(x) \geq 0$$

where $f : \mathbb{R}^n \to \mathbb{R}, c : \mathbb{R}^m \to \mathbb{R}^n$ *where* $c = [c_1, ..., c_m]^T, c_i : \mathbb{R}^n \to R$.

Definition 8.4.2. *Consider a x satisifying a constraint $c_i(x) \geq 0$. We said*

- *the constraint is **active or binding** at x_0 if $c(x_0) = 0$;*
- *the constraint is **inactive or not binding** at x_0 if $c(x_0) > 0$.*

Definition 8.4.3 (feasible path, tangent vector). *[1, lec 4]A feasible path is a curve for constraints $c(x) = 0$, represented by a twice continuously differentiable function $x(t)$, that emanates from a feasible point x_0 such that*

$$x(0) = x_0, c(x(t)) = 0$$

for all $0 \leq t < \sigma, \sigma > 0$, and such that $dx/dt|_{t=0} \neq 0$ The tangent vector of the feasible path is given as

$$p = dx(t)/dt|_{t=0}.$$

Definition 8.4.4 (tangent cone). *[1, lec 4]Given constraints $c(x) \geq 0$, the set*

$$\mathcal{T}(x) = \{p : p \text{ is a nonzero vector tangent to a feasible path emanating from } x\} \cup \{0\}\}$$

is called the tangent cone of c at the point x.

Theorem 8.4.1 (first order necessary condition, geometric form). *[1, lec 4]If x^* is a local minimizer for inequality constraint optimization problem, then*

$$c(x^*) \geq 0 \text{ and } \nabla f(x^*)^T p \geq 0, \forall p \in \mathcal{T}(x^*)$$

.

Proof. Suppose for some p, we have $\nabla f(x^*)^T p < 0$, then in direction p, let $\gamma(t)$ be the feasible curve emanating from x^* with tangent vector p, we have

$$f(\gamma(\alpha)) = f(x^*) + \alpha f(x^*)^T p + O(\alpha^2) < f(x^*)$$

as $\alpha \to 0, \alpha > 0$, which contradicts that fact that $f(x^*)$ is local minimum. □

Remark 8.4.1 (the need for algebraic condition). Note that this theorem gives a necessary condition in geometric form, which is not easy to use in practice.(for example, the set $\mathcal{T}(x^*)$ is not explicit.) We need to convert these geometric condition to algebraic conditions for the ease of use. These algebraic condition is known as **KKT** condition.

8.4.2 Constraint qualifications and KKT conditions

Definition 8.4.5 (linearly feasible direction). *Given a feasible point x for the constraint $c(x) \geq 0$ and $\mathcal{A} = \mathcal{A}(x)$, we define the set of* ***linearly feasible directions*** *as*

$$\mathcal{T}_L(x) = \{p : J_{\mathcal{A}}(x)p \geq 0\}$$

Remark 8.4.2 (interpretation on linearly feasible). Let x_0 be a feasible point, and let $\mathcal{A}$ denote the active constraint set. Then $c_i(x_0) = 0, \forall i \in \mathcal{A}$. Let $p \in \mathcal{T}_L$, then for sufficiently small $\alpha > 0$, we have

$$c_i(x_0 + \alpha p) \approx c_i(x_0) + \alpha \nabla c_i^T p \geq 0.$$

That is, $x_0 + \alpha p$ is still in feasible region, and p is a feasible direction.

Lemma 8.4.1.

$$\mathcal{T}(x) \subseteq \mathcal{T}_L(x)$$

Proof. Let $r(t)$ be a feasible path for c_i, then $c_i(r(t)) \geq 0$. Take the derivative repsect to r at $t = 0$, we have

$$\nabla c_i \cdot \frac{dr}{dt}|_{t=0} = \nabla c_i \cdot p \geq 0$$

Therefore, if $p \in \mathcal{T}(x)$, we have $\nabla c_i \cdot p \geq 0, i \in \mathcal{A}$ and implies $J_A^T p \geq 0$. □

Definition 8.4.6 (constraint qualification). *We say that the constraint qualification of inequality constraint $c(x) \geq 0$ holds at a feasible point x if every nonzero vector p satisfying $J(x)_A^T p = 0$ implies $p \in \mathcal{T}_L(x)$.*

Remark 8.4.3 (purpose of constraint qualifications).

- constraint qualifications is simple the assumptions on constraint such that later KKT condition relies on
- There are different types of constraint qualifications, such as linear independence constraint qualification, Managasarian-Fromovitz constraint qualification.

Lemma 8.4.2. *If constraint qualifications of inequality constraint $c(x) \geq 0$ holds, then*

$$\mathcal{T}(x) = \mathcal{T}_L(x)$$

Proof. Directly from above lemma $\mathcal{T}(x) \subseteq \mathcal{T}_L(x)$ and the definition and constraint qualification. □

Lemma 8.4.3 (linear constraints always satisfy constraint qualifications). *The constraint qualification of equality constraint $c(x) \geq 0$ holds at x if $c(x) = Ax - b \geq 0$(no matter A is full row rank or not.)*

Proof. Let A_a denote the constraint matrix consists of active constraints. Denote x_0 as the feasible point. Then $A_a x_0 = b_a$. $A_a(x_0 + \alpha p) = b_a$ for sufficiently small $\alpha > 0$ and $p \in \mathcal{N}(A_a)$. Based on the definition, it satisfies the constraint qualification. □

Lemma 8.4.4 (sufficient condition: linear independence constraint qualifications, nonlinear constraint case). *[2, p. 324] The constraint qualification of inequality constraint $c(x) \geq 0$ holds at x if $J_A(x)$ has full row rank(i.e. $\nabla c_i, i \in \mathcal{A}$ are linearly independent of each other). And this is called* ***linear independence constraint qualification****.*

Lemma 8.4.5 (Farkas's lemma, alternative). *Let $g \in \mathbb{R}^n$ and $A \in \mathbb{R}^{r\times n}$. It follows that*

$$g^T p \geq 0, \forall p \in \{p : Ap \geq 0\}$$

if and only if there exists $\lambda \in \mathbb{R}^r, \lambda \geq 0$ such that

$$g = A^T \lambda.$$

Proof. (1) forward. If $g = A^T\lambda$, then $g^T p = \lambda A p \geq 0$; (2) converse. If $g \notin \mathcal{X} = \{A^T\lambda, \lambda \geq 0\}$, then g is the point lying outside the cone $\mathcal{X}$. Based on Theorem 10.2.3, there exists p such that $Ap \geq 0$(all the basis vectors of the cone lying on one halfspace of a hyperplane passing origin and having norm vector p) and $g^T p < 0$(the element g lying on the other halfspace). This contradicts that for all p such that $Ap \geq 0$, $g^T p \geq 0$. □

Theorem 8.4.2 (first order condition for inequality constraints, KKT condition). *Assume that the first-order constraint qualification holds. If x^* is a local solution to nonlinear optimization with inequality constraint [Definition 8.4.1), then there exists a Lagrange multiplier vector y^* such that*

$$\begin{aligned} c(x^*) &\geq 0 \ (feasiblity) \\ g(x^*) &= \sum_{i\in\mathcal{A}} [y^*]_i \nabla c_i(x^*), [y^*]_i \geq 0, \forall i \in \mathcal{A} \\ [y^*]_i &= 0, \forall i \in \mathcal{I} \end{aligned}$$

Or equivalently,

$$\begin{aligned} c(x^*) &\geq 0 \\ g(x^*) &= J(x^*)^T y^*, y^* \geq 0 \\ c_i(x^*) y_i^* &= 0, \forall i \end{aligned}$$

Proof. When the constraint qualification holds, we have $\mathcal{T}(x^*) = \mathcal{T}_L(x^*) = \{p : J_{\mathcal{A}}(x^*)p \geq 0\}$. The optimality condition [Theorem 8.4.1] says if x^* is a local solution, then $c(x^*) \geq 0$ and $g^T(x^*)p \geq 0, \forall p \in \mathcal{T}_L(x^*) = \{p : J_{\mathcal{A}}(x^*)p \geq 0\}$. The Farkas' lemma [Lemma 8.4.5] says that the gradient at x^* must satisfy

$$g(x^*) = J_{\mathcal{A}}^T y^*, y_{\mathcal{A}}^* \geq 0.$$

Further considering inactive constraints, we can view them as non-existence. So we directly set $y_{\mathcal{I}}^* = 0$. □

Remark 8.4.4 (comparison with equality-constraint optimization).

- In equality constraint problem, there is no constraints on the multipliers' value; in inequality constraint problem, we require the the multipliers' value to be nonnegative.
- The equality constraint problem can be constructed from inequality constraint problems by pairing constraints, i.e. $c(x) = 0 \Leftrightarrow c(x) \geq 0, -c(x) \geq 0$. Then, Theorem 8.4.2 will reduce to

Note 8.4.1 (interpretation of Lagrange multipliers in inequality constraints). [1, lec 4] Assume (x^*, y^*) is a first-order KKT point and $J_{\mathcal{A}}(x^*)$ has full row rank. Let p_i be a vector(such p_i will always exists since $J_{\mathcal{A}}$ is full row rank) such that

$$J_{\mathcal{A}} p_i = e_i, \forall i = 1, ..., |\mathcal{A}|,$$

where e_i is the ith coordinate basis vector. Then moving in the direction p_i will step off the ith active constraint, but still tangent to the other active constraints.

Observe that

$$g(x^*)^T p_i = (y^*_{\mathcal{A}})^T J_{\mathcal{A}}(x^*) p_i = (y^*_{\mathcal{A}})^T e_i = [y^*_{\mathcal{A}}]_i.$$

Therefore, if $[y^*_{\mathcal{A}}]_i > 0$, then moving in the direction p_i will result in initial increasing of the objective function; vice versa. If $[y^*_{\mathcal{A}}]_i = 0$, then moving in the direction p_i will not result in initial change.

In summary,

- A positive Lagrange multiplier associated with an active constraint implies that the objective function initially increases when stepping off of that constraint.
- A zero Lagrange multiplier associated with an active constraint implies that the objective function f initially is 'flat' when stepping off of that constraint. This further implies the minimizer is not strict.
- A negative Lagrange multiplier associated with an active constraint implies that the objective function f initially is decreasing when stepping off of that constraint. This further implies this is not a minimizer.
- If a constraint is inactive, then its Lagrange multipliers must be zero.
- We require $y^* \geq 0$ indicates any deviation of current point will result initial non-decreasing.

Note 8.4.2 (geometry of moving with inequality constraints). Consider a direction vector $p \in \mathbb{R}^n$. Let i denote an active inequality constraint at a feasible x.

- If $\nabla c_i(x)^T p = 0$, then $c_i(x + \alpha\alpha p) \approx c_i(x) + \alpha \nabla c_i(x)^T p = 0$.
- If $\nabla c_i(x)^T p < 0$, then $c_i(x + \alpha\alpha p) \approx c_i(x) + \alpha \nabla c_i(x)^T p < 0$.
- If $\nabla c_i(x)^T p > 0$, then $c_i(x + \alpha\alpha p) \approx c_i(x) + \alpha \nabla c_i(x)^T p > 0$.

8.4.3 Second order conditions

Notations:

$$\mathcal{A}_+(x^*) = \mathcal{A}(x^*) \cap \{i : y_i^* > 0\}$$
$$\mathcal{A}_0(x^*) = \mathcal{A}(x^*) \cap \{i : y_i^* = 0\}$$

Remark 8.4.5 (general remarks on second order condition).

- Second order condition will examine curvature effect along paths for which f is initially 'flat'.
- Simple examples include $f(x) = x^3$ and $f(x) = x^4$. Only second order condition can help distinguish optimality at $x = 0$.
- If $y_i^* > 0$, then from 8.4.1 we know that stepping off i constraint will increase objective function; however, move along the constraint surface will not.
- If $y_i^* = 0$ for an active constraint, stepping off i constraint or move along i will be flat.

Definition 8.4.7 (initial flat linearized direction set).

$$\mathcal{S} = \{p : \nabla c_i(x^*)^T p = 0, \forall i \in \mathcal{A}_+(x^*) \cup \nabla c_i(x^*)^T p \geq 0, \forall i \in \mathcal{A}_0(x^*)\}$$

Remark 8.4.6. If $\nabla c_i(x^*)^T p = 0$, then move along p will maintain the acitve constraint i active; If $\nabla c_i(x^*)^T p > 0$, then move along p will step off the acitve constraint i into feasible region; If $\nabla c_i(x^*)^T p < 0$, then move along p will step off the acitve constraint i into infeasible region.

Lemma 8.4.6. *For any $p \in \mathcal{S}$,*

$$g^T p = 0.$$

Proof. Note that

$$g(x^*)^T p = \sum_{i \in \mathcal{A}} y_i^* (\nabla c_i(x^*))^T p.$$

If $y_i^* > 0$, we require $\nabla c_i(x^*)^T p = 0$. If $y_i^* = 0$, we relax to $\nabla c_i(x^*)^T p \geq 0$. □

Definition 8.4.8 (second order constraint qualification for inequality problems). *The second order constraint qualification for inequality constraint $c(x) \geq 0$ holds at a KKT poyint (x^*, y^*) if for all y in*

$$\mathcal{M}(x^*) \triangleq \{y : g(x^*) = J(x^*)^T y, y \geq 0 \text{ and } c_i(x^*) * y_i = 0\}$$

It follows that every nonzero direction p in the set

$$\mathcal{S} = \{p : \nabla c_i(x^*)^T p = 0, \forall i \in \mathcal{A}_+(x^*) \cup \nabla c_i(x^*)^T p = 0, \forall i \in \mathcal{A}_0(x^*)\}$$

is tangent to a twice-differentiable path $x(\alpha)$ such that $c_{\mathcal{A}_+}(x(\alpha)) = 0$ and $c_{\mathcal{A}_0}(x(\alpha)) \geq 0$ for all $0 < \alpha \leq \sigma$ and some $\sigma > 0$.

Remark 8.4.7 (interpretation)**.** The set of twice-differentiable path $x(\alpha)$ such that $c_{\mathcal{A}_+}(x(\alpha)) = 0$ and $c_{\mathcal{A}_0}(x(\alpha)) \geq 0$ for all $0 < \alpha \leq \sigma$ and some $\sigma > 0$ **is a set of path that initially flat and feasible**. Usually, the tangent vector set of this set of paths might be different from $\mathcal{S}$. The constraint qualification says that they are equivalent.

Remark 8.4.8 (comparison with first order constraint qualification)**.**

- The first order constraint qualification does not imply the second order constraint qualification.
- The second order constraint qualification does not imply the first order constraint qualification.
- If active constraints are linear, then both first and second order constraint qualification hold.
- Linear independence qualification is sufficient for both first and second order constraint qualification.

Theorem 8.4.3 (second-order necessary conditions). *[1, lec 4] Suppose that $f, c \in C^2$. If x^* is local constrained minimizer of the inequality constrained problem at which the first and the second order constraint qualifications are satisfied, then there exists a vector of Lagrange multiplier y^* such that*

- $c(x^*) \geq 0$
- $g(x^*) = J(x^*)^T y^*, y^* \geq 0$
- $c(x^*) \cdot y^* = 0$*(complementary slackness)*
- $$p^T[\sum_{i=1}^m -\lambda_i^* \nabla^2 c_i(x^*) + H(x^*)]p \geq 0$$

  holds for all $p \in \mathcal{S}$.

Theorem 8.4.4 (second-order sufficient condition). *[1, lec 4] Suppose that $f, c \in C^2$. If there exist (x^*, y^*) such that*

- $c(x^*) \geq 0$
- $g(x^*) = J(x^*)^T y^*, y^* \geq 0$
- $c(x^*) \cdot y^* = 0$ *(complementary slackness)*
- $$p^T [\sum_{i=1}^m -\lambda_i^* \nabla^2 c_i(x^*) + H(x^*)] p \geq \omega \|p\|$$
  for some $\omega > 0$ and for all $p \in \mathcal{S}$.

Then, x^* is local constrained minimizer.

Example 8.4.1. Consider a trust region subproblem [Theorem 7.3.1]

$$\min_{s \in \mathbb{R}^n} m(s) = f + s^T g + \frac{1}{2} s^T B s, \text{ subject to } \|s\|_2 \leq \delta.$$

Note that we can write the trust region constraint as $-\frac{1}{2} s^T s \geq -\frac{1}{2}\delta^2$, whose $J(s) = -s$

A vector s^* is a local minimizer if and only if $\|s^*\| \leq \delta$ and there exists a scalar $\lambda^* \geq 0$ such that

- $\|s^*\| \leq \delta$.
- $(g + Bs) = -\lambda^* s^*$ gives $(B + \lambda^* I)s^* = -g$.
- (complementary slackness)$\lambda^*(\|s\|_2 - \delta) = 0$
- $B + \lambda^* I$ is positive definite.

Definition 8.4.9 (strict complementarity). *We say that strict complementarity holds at the KKT point x^* if there exists a Lagrange multiplier vector y^* such that $y^* > 0$ for all $i \in \mathcal{A}(x^*)$.*

Remark 8.4.9 (why require strict complementarity). If a Lagrange multiplier y_i associated with an active constraint is zero, then it implies the minimizer is not strict. See 8.4.1.

Corollary 8.4.4.1 (second-order sufficient condition). *[1, lec 4] Suppose that $f, c \in C^2$. If there exist (x^*, y^*) such that*

- $c(x^*) \geq 0$
- $g(x^*) = J(x^*)^T y^*, y^* \geq 0$

- *$c(x^*) \cdot y^* = 0$(complementary slackness)*
- ***strict complementarity is satisfied**.*
- $$p^T[\sum_{i=1}^{m} -\lambda_i^* \nabla^2 c_i(x^*) + H(x^*)]p \geq \omega \|p\|_2^2$$

  for some $\omega > 0$ and for all $p \in \mathcal{N}(J(x^*))$.

Then, x^* is local constrained minimizer.

Proof. Based on the definition, strict complementarity means that $\mathcal{A}_0 = \emptyset$ and $\mathcal{S} = \mathcal{N}(J_{\mathcal{A}}(x^*))$. $\square$

8.5 Envelope theorem and sensitive analysis

Lemma 8.5.1 (sensitivity of unconstrained optimization). *[5, p. 369] Consider a maximization problem given by*

$$M(a) = \max_{x \in \mathbb{R}^n} f(x,a)$$

where a is an external parameter. Denote the maximizers of f as $x(a)$.Assume $M(a)$ and $x(a)$ are differentiable with respect to a. Then

$$\frac{dM(a)}{da} = \frac{\partial f(x,a)}{\partial a}|_{x=x(a)}.$$

Proof.

$$\frac{dM(a)}{da} = \sum_{i=1}^{n} \frac{\partial f(x,a)}{\partial x_i} \frac{dx_i(a)}{da} + \frac{\partial f(x,a)}{\partial a}.$$

Note that the first term of the right-hand-side is zero due to first order necessary condition of unconstrained optimization

$$\frac{\partial f}{\partial x_i} = 0, i = 1,2,...,n.$$

□

Lemma 8.5.2 (sensitivity of equality constrained optimization,envelope theorem). *[5, p. 369][6, p. 605] Consider a maximization problem given by*

$$M(a) = \max_{x \in \mathbb{R}^n} f(x,a),\ s.t.\ g_i(x,a) = 0, j = 1,2,...,m$$

where a is an external parameter. Denote the maximizers of f as $(x(a), \lambda(a))$. Denote the Lagrange function as

$$L(x,\lambda,a) = f(x,a) - \sum_{j=1}^{m} \lambda_j g_j(x,a).$$

Assume $M(a), x(a), \lambda(a)$ are differentiable with respect to a. Then

$$\frac{dM(a)}{da} = \frac{\partial L(x,\lambda,a)}{\partial a}|_{x=x(a),\lambda=\lambda(a)}.$$

Proof.

$$\frac{dM(a)}{da} = \nabla_x f(x,a) \cdot \frac{dx}{da} + \frac{\partial f(x,a)}{\partial a}.$$

Use the first order condition on the gradient, we have

$$\nabla_x f(x,a) = \sum_{j=1}^{m} \lambda_j \nabla_x g_j(x(a),a).$$

Use the feasibility condition $g_j(x(a),a) = 0, j = 1,2,...,m$ and take derivative with respect to a, we have

$$\nabla_x g_j \cdot \frac{dx}{da} + \frac{\partial g_j}{\partial a} = 0.$$

Therefore, we have

$$\frac{dM(a)}{da} = -\sum_{j=1}^{m} \lambda_j \frac{\partial g_j(x,a)}{\partial a} + \frac{\partial f(x,a)}{\partial a} = \frac{\partial L(x,\lambda,a)}{\partial a}.$$

□

Remark 8.5.1 (same conclusion for minimization problem)**.** For the minimization problem, the same conclusion will hold. Note in the proof we do not distinguish whether we are minimizing or maximizing.

Corollary 8.5.0.1 (sensitivity of equality constrained optimization). *[5, p. 369][6, p. 605] Consider a maximization problem given by*

$$M(a) = \max_{x\in\mathbb{R}^n} f(x),\ s.t.\ g_i(x) = a_i, j = 1,2,...,m$$

where a is an external parameter. Denote the maximizers of f as $(x(a),\lambda(a))$. Denote the Lagrange function as

$$L(x,\lambda,a) = f(x,a) - \sum_{j=1}^{m} \lambda_j g_j(x,a).$$

Assume $M(a), x(a), \lambda(a)$ are differentiable with respect to a. Then

$$\frac{\partial M(a)}{\partial a} = \lambda_i.$$

Proof. Note that

$$L(x,\lambda,a) = f(x) - \sum_{i=1}^{m} \lambda_i (g_i(x) - a_i) \implies \partial L/\ Paa_i = \lambda_i.$$

□

Lemma 8.5.3 (sensitivity of inequality constrained optimization). *[5, p. 369][6, p. 605] Consider a maximization problem given by*

$$M(a) = \max_{x \in \mathbb{R}^n} f(x,a), \text{ s.t. } g_i(x,a) \geq 0, j = 1,2,...,m$$

where a is an external parameter. Denote the maximizers of f as $(x(a), \lambda(a))$. Denote the Lagrange function as

$$L(x,\lambda,a) = f(x,a) - \sum_{j=1}^{m} \lambda_j g_j(x,a).$$

Assume $M(a), x(a), \lambda(a)$ are differentiable with respect to a. Then

$$\frac{dM(a)}{da} = \frac{\partial L(x,\lambda,a)}{\partial a}|_{x=x(a),\lambda=\lambda(a)}.$$

Proof. Note that at optimality, λ_i is 0 for inactive constraints. Therefore, we should only consider the problem as an equality constrained problem [Lemma 8.5.2]. □

8.6 Notes on bibliography

For introductory level of optimization theory, see [7].

For intermediate treatment, see [1][2].

BIBLIOGRAPHY

1. Robinson, D. *Nonlinear optimization II lecture notes* (Johns Hopkins University, 2015).
2. Nocedal, J. & Wright, S. *Numerical optimization* (Springer Science & Business Media, 2006).
3. Bertsekas, D. *Nonlinear programming* ISBN: 9781886529007 (Athena Scientific, 2016).
4. Robinson, D. *Nonlinear optimization I lecture notes* (Johns Hopkins University, 2015).
5. Jiang, D. *advanced microeconomics (chineses)* (Peking University Publisher, 2000).
6. Jehle, G. & Reny, P. *Advanced Microeconomic Theory* ISBN: 9780273731917 (Financial Times/Prentice Hall, 2011).
7. Sundaram, R. K. *A first course in optimization theory* (Cambridge university press, 1996).

9

LINEAR OPTIMIZATION

9.1 Equality constrained linear programming

In equality constrained linear programming, we aim to solve following optimization problems,

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c_1x_1 + \cdots + c_nx_n \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m. \end{aligned}$$

Writing compactly in matrix form, we have

Definition 9.1.1 (equality constrained linear programming). *A equality constrained linear programming is given as:*

$$\min_{x\in\mathbb{R}^n} c^T x$$
$$\text{subject to } Ax = b$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $rank(A) = m < n$, and $Ax = b$ is assumed to be consistent.

Remark 9.1.1.

- **Assuming** $rank(A) = m$ **will not cause lose of generality**, since if we can always make A full rank by removing linear dependent rows.
- **The situation** $m \geq n$ **is trivial**: either we have only one feasible solution or we do not have any feasible solution.
- **Assuming** $Ax = b$ **is consistent** will ensure feasible region is not empty.

Under constraint $Ax = b$, the feasible region is a point x_0 satisfying $Ax_0 = b$ plus the null space of A, denoted by $\mathcal{N}(A)$. If c is not perpendicular to $\mathcal{N}(A)$, then the objective function $c^T x$ can take value from $-\infty$ to ∞. Otherwise, the objective function will take a constant value within the whole feasible region.

We have the following summary.

Theorem 9.1.1 (solution to equality constrained linear programming). *Suppose $rank(A) = m < n$ and $Ax = b$ is consistent. The solution to the equality constrained linear programming problem has exactly two possibilities:*

1. *If $Ax = b$ is consistent and $c \perp \mathcal{N}(A)$, then every solution to $Ax = b$ is a minimizer; moreover, these minimizers are all equal and they are all global minimizers.*
2. *If $Ax = b$ is consistent and $c \not\perp \mathcal{N}(A)$, then the optimization problem is unbounded blow.*

Moreover, check $c \perp \mathcal{N}(A)$ is equivalent to whether $c \in \mathcal{R}(A^T)$ or the consistence of linear equation $c = A^T z$.

Proof. (1) We can decompose every feasible solution $x = x_0 + y$ uniquely, where y is a vector in the null space of $A(y \in \mathcal{N}(A))$, x_0 is a solution to $Ax_0 = b$ and $x_0 \perp \mathcal{N}(A), x_0 \in \mathcal{R}(A^T)$(note that $\mathcal{R}(A^T) \perp \mathcal{N}(A)$ by the fundamental theorem of linear algebra). Then we have minimum $c^T x = c^T x_0 + c^T y = c^T x_0$ because $c \perp \mathcal{N}(A)$, with x_0 being the minimizer. Other feasible solution can be written as $x + w$, where $w \in \mathcal{N}(A)$ and $c^T w = 0$; (2) The objective function can be written as $c^T x = c^T x_0 + c^T y$, which can be made to $-\infty$, since y can be an arbitray vector in the null space of A;

(3) Since $\mathcal{R}(A^T) \perp \mathcal{N}(A), \mathcal{N}(A)^{\perp} = \mathcal{R}(A^T)$, from fundamental theorem of linear algebra. $\square$

9.2 Inequality constrained linear programming

9.2.1 Linear optimization with inequality constraints

In equality constrained linear programming, we aim to solve following optimization problems,

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c_1x_1 + \cdots + c_nx_n \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m. \end{aligned}$$

Writing compactly in matrix form, we have

Definition 9.2.1 (canonical form inequality constrained linear optimization). *A linear programming canonical form is:*

$$\begin{aligned} & \textit{minimize } c^T x \\ & \textit{subject to } Ax \geq b \end{aligned}$$

*where $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n$. **And we require** $rank(A) = n \leq m$ **(tall and thin)***

Remark 9.2.1 (generality of canonical form)**.** The canonical form covers both equality and inequality constraints. To see this, consider an equality constraint $\{a_i^T x = b_i, i = 1, ..., q\}$, we can convert it to inequality constraints as $\{a_i^T x \geq b_i, -a_i^T x \leq -b_i\}$

Remark 9.2.2 (Why we require $rank(A) = n$ and $m \geq n$)**.**

- If $rank(A) < n$, then $Ax = b$ has the solution being an affine set. If c is not perpendicular to the $\mathcal{N}(A)$, then $c^T x$ will be unbounded(then the minimizer might not exist).
- If $m < n$, then $rank(A) \leq m < n$, we end up with the first situation.

9.2.2 Geometry of linear programming

The feasible set can be viewed the intersection of finite half spaces. When moving in the $-c$ direction in the feasible region, the objective function decreases. There are several critical observations.

- The feasible region is an open space extending to infinity if A is not full column rank. In this case, the objective function is usually unbounded below unless c is perpendicular to $\mathcal{N}(A)$[Figure 9.2.1(a)].
- Under the assumption $rank(A) = n$ and $m \geq n$, the feasible set is either a 'close' polyhedron [Figure 9.2.1(a)(b)]. or a 'half-close-half-open' polyhedron [Figure 9.2.1(d)].
- If the feasible region is a 'close' polyhedron, the minimizer could be found by tracking down along $-c$ until hitting an extreme vertex.
- If the feasible region is a 'half-close-half-open' polyhedron, the objective function could be unbounded below if $-c$ pointing outward.

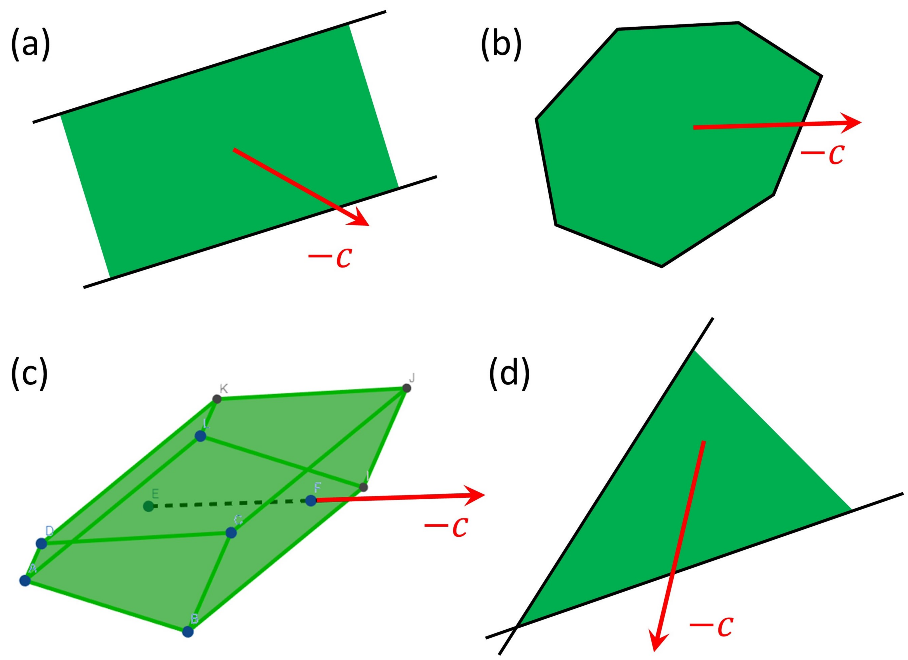


Figure 9.2.1: The geometry of linear programming. (a) The feasible region is an open space extending to infinity if A is not full column rank. (b-d) Example feasible regions if $rank(A) = n, m \geq n$. Red arrows are direction of $-c$. When moving along $-c$ in the feasible region, the objective function will decrease.

Example 9.2.1. Consider a company that produces product A and B. A and B can be sold at prices of \$8 and \$10, respectly. The cost to produce A and B are \$4 and \$7. The production budget is \$100. Let x and y denote the quantity of A and B to produce. Then the goal to maximize profit is given by the following linear program

$$\begin{cases} \max & f(x,y) = 4x + 3y \\ \text{subject to} & 4x + 7y \leq 100 \\ & y \geq 0 \\ & x \geq 0 \end{cases}$$

9.2.3 Optimality property and condition

Lemma 9.2.1 (local minimum implies global minimum). *Let x^* be a local minimizer. Then x^* is also a global minimizer.*

Proof. let x^* be a local minimizer, suppose there is a point $x' \neq x^*$ being a global minimum such that $c^T x' < c^T x^*$. Then

$$c^T(x^* + \alpha(x' - x^*)) = \alpha c^T x^* + (1-\alpha)c^T x' < \alpha c^T x^* + (1-\alpha)c^T x^* < c^T x^*$$

for $\alpha \in [0,1]$. Therefore, starting from $c^T x^*$, if we move along $(x' - x^*)$ for a sufficiently small step, the objective function value will decrease. This contradicts that x^* is local minimizer. □

Theorem 9.2.1 (first-order KKT conditions are sufficient and necessary). *[1, lec 6] If there exists a KKT point, i.e., a point $(x^*, \lambda_a), x^* \in \mathbb{R}^n$ that satisfies*

$$Ax^* \geq b, c = A_a^T y_a, y_a \geq 0,$$

where A_a is the active constraint matrix at x^*, then (x^*, λ_a) is a minimizer.

Moreover, the objective function is unbounded below on the feasible region if and only if there does not exists a $y_a \geq 0$ such that $c = A_a^T y_a$.

Proof. (1) The first order KKT necessary condition directly from Theorem 8.4.2Theorem 8.3.2. We have the kkT condition on (x^*, y^*) given by,

$$\begin{aligned} Ax^* \geq b, x^* \in \mathbb{R}^n (primal feasibility) \\ A^T y^* = c, y^* \geq 0, y^* \in \mathbb{R}^m (dual feasibility) \\ [Ax^* - b]_i [y^*]_i = 0, \forall i = 1, ..., m (complementarity) \end{aligned}$$

When a constraint s is inactive, then $[y^*]_s = 0$, therefore $A^T y^* = c$ can be simplified as $c = A_a^T y_a$. For active constraints, note that the complementary conditions for these active index are automatically satisfied. (2)(sufficiency) Note that since $y^* \geq 0, Ax^* \geq b$, the complementary condition is also equivalent to

$$(Ax^* - b)^T y^* = 0 \Leftrightarrow b^T y^* = c^T x^*.$$

Using this result, we can show the KKT condition is also sufficient: Let $\bar{x}$ be any other feasible point such that $A\bar{x} \geq b$. Then

$$c^T \bar{x} = (A^T y^*)^T \bar{x} = (A\bar{x})^T y^* \geq b^T y^* = c^T x^*,$$

where we used the fact that $A\bar{x} \geq b, y^* \geq 0$. □

Lemma 9.2.2 (uniqueness of minimizer, affine geometry of minimizers). *If the linear optimization problem has a minimizer, then there are* ***exactly*** *two possibilities:*

- *it has a unique minimizer.*
- *it has infinitely many minimizers forming an convex hull; or equivalently, if it has more than one minimizers, then it has infinitely many minimizers.*

Proof. Given two minimizers (x_1, y_1) and (x_2, y_2), then it is easy to see that the convex combination of the two will still satisfy the KKT condition [Theorem 9.2.1]. It can be generalized to multiple minimizers. □

Example 9.2.2. Consider the linear programming problem illustrated in Figure 9.2.2. The red edge that normal to c are all minimizers.

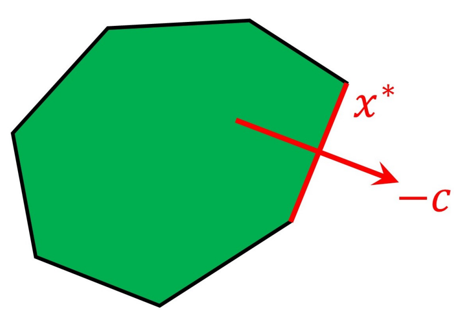


Figure 9.2.2: Demonstration on multiple minimizers forming a convex set.

9.2.4 Standard form of linear programming

Definition 9.2.2 (standard form linear optimization). *[2, p. 4] A linear programming standard form is:*

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Ax = b \\ &\qquad\qquad x \geq 0 \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n$

Reductions to standard forms from canonical form in two steps:

- For an unrestricted x_i, replaced with $x = x_i^+ - x_i^-$ and add constraint $x_i^+ \geq 0, x_i^- \geq 0$
- For an inequality constraint $a^T x \geq b_i$, replaced with

$$a^T x - s = b$$

$$s \geq 0$$

9.2.5 Application examples

Lemma 9.2.3. *The optimization problem*

$$\min_x [\max_{1\leq j\leq k} \{l_j(x)\}]$$

where $x \in \mathbb{R}^n, l_j(x) = a_j^T x + b_j, a_j \in \mathbb{R}^n, b_j \in \mathbb{R}$ *is equivalent to*

$$\min_{x,v} v, \ s.t. \ l_j(x) \leq v, \forall 1 \leq j \leq k.$$

Proof. Note that the constraints

$$l_j(x) \leq v, \forall 1 \leq j \leq k, \forall x, v,$$

implies

$$v \geq \max_{1\leq j\leq k} l_j(x), \forall v, x$$

Note that left side is a function of v and right side is a function of x. We therefore minimize both sides on v and x simultaneously and we get

$$\min_{v,x} v = \min_x [\max_{1\leq j\leq k} \{l_j(x)\}].$$

□

9.3 Linear programming geometry and simplex algorithm

9.3.1 Geometrical approach to linear programming

9.3.1.1 *Overview*

Following the initial discussion on the geometry of linear programming [subsection 9.2.2], we now aim to develop an algorithm based on the geometry. Given a bounded feasible region [Figure 9.3.1(a)], an intuitive way to search for the extreme vertex that minimize the objective function is to move from one vertex to another along edges that would decrease objective function values. For a bounded feasible region [Figure 9.3.1(b)], we would either find a minimizer or find out if the objective function is unbounded below.

Based on this insight, a complete algorithm should consist the following steps:

- Determine if a vertex is local minimizer or not.
- If a vertex is not the minimizer, determine which edge or direction to move to.
- Determine the step size to move to exactly a new vertex.

These steps are indeed the essence of the famous Simplex algorithm. In the following sections, we will examine how to mathematically formulate the procedures for each step.

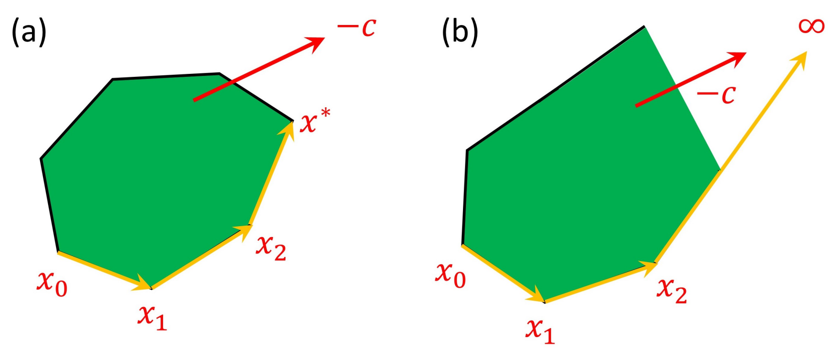


Figure 9.3.1: Overview of geometry approach to linear programming.

9.3.1.2 *Vertex and optimality*

Definition 9.3.1 (vertex, degeneracy). *Given a set of m linear constraints in n variables, a vertex is a feasible point for which the active-constraint matrix contains* ***at least one subset of*** n ***linearly independent rows****, i.e.,* $rank(A_a) = n$.

*If **exactly** n constraints are active at a vertex, the vertex is said to be **non-degenerate**. If more than n constraints are active at a vertex, then it is said to be **degenerate**.*

Remark 9.3.1 (interpretation).

- **Any vertex $x^* \in \mathbb{R}^n$ is a single point**, since $A_a x^* = b_a$ has an unique solution when $rank(A_a) = n$.
- A vertex can only exist when $m \geq n$. Algebraically, if $m < n$, and $rank(A_a) \leq m < n$. Geometrically, if $m < n$, then $Ax = 0$ has infinite solutions, and solution to $Ax* = b$ will a affine plane instead of a point.

Definition 9.3.2 (adjacent vertex). *Let x_1 and x_2 be two vertices of the feasible region $\mathcal{F}$ for the constraints $Ax \geq b$, and let A_1 and A_2 denote their associated active-constraint matrices. We say that x_1 and x_2 are **adjacent vertices** if the matrix B formed from the rows common to both A_1 and A_2 has $rank(B) = n - 1$.*

*In particular, if x_1 and x_2 are both **non-degenerate adjacent vertices**, then A_1 and A_2 has rank $n - 1$.*

Lemma 9.3.1 (existence of a vertex). *Consider the constraints $Ax \geq b$, where $A \in \mathbb{R}^{m \times n}$. If*

- *there exists at least one feasible point;*
- *$rank(A) = n$*

then a vertex exists.

Proof. Suppose x_0 is a feasible but not a vertex. Let $A_{a(x_0)}$ be the active constraint matrix at x_0. Since x_0 is not vertex, then $dim(\mathcal{N}(A_{a(x_0)})) > 0$. Therefore, we can select $p \in \mathbb{R}^n, p \in \mathcal{N}(A_{a(x_0)})$ such that $x_1 = x_0 + \alpha p, \alpha \in \mathbb{R}$ will still satisfy the original $A_{a(x_0)} x_1 = b_{a(x_0)}$ since $A_{a(x_0)} p = 0$. More importantly, we can select α such that we activate one more constraint. (such α must exist since our feasible region does not contain an affine set(see subsection 9.2.2]).

Therefore, as we continue the process, as long as x_k is not a vertex such that $dim(\mathcal{N}(A_{a(x_k)})) = 0$, we can always move to activate more constraint until we hit a vertex. □

Theorem 9.3.1 (existence of a vertex minimizer). *[1, lec 6] Consider the linear programming [Definition 9.2.1]. If*

- *at least one feasible point exists*
- *$rank(A) = n$*

- *the objective function is bounded blow on the feasible region (or equivalently, the objective function has a minimizer on the feasible region)*

*Then, there will **exist a vertex being the minimizer.***

Proof. Suppose x_0 is a minimizer but not a vertex. Let $A_{a(x_0)}$ be the active constraint matrix at x_0. Since x_0 is not vertex, then $dim(\mathcal{N}(A_{a(x_0)})) > 0$. Therefore, we can select $p \in \mathbb{R}^n, p \in \mathcal{N}(A_{a(x_0)})$ such that $x_1 = x_0 + \alpha p, \alpha \in \mathbb{R}$ will still satisfy the KKT condition Theorem 9.2.1 and will not change of value of the objective function since $c^T p = y_a^T A_a p = 0$. More importantly, we can select α such that we activate one more constraint. (such α must exist since our feasible region does not contain an affine set(see subsection 9.2.2).

Therefore, as we continue the process, as long as x_k is not a vertex such that $dim(\mathcal{N}(A_{a(x_k)})) = 0$, we can always move to activate more constraint until we hit a vertex. □

Remark 9.3.2 (implications on optimization algorithms).

- If the feasible set is not empty, then the vertex existence lemma [Lemma 9.3.1] guarantees that there are vertices in the feasible set.
- Further, the vertex minimizer theorem [Theorem 9.3.1] guarantees that among these feasible vertices, one of them must be a minimizer.
- Therefore, we can either enumerate all vertices either by brute force or by smarter vertices search (simplex algorithm).

Theorem 9.3.2 (uniqueness of minimizer and vertex minimizer, non-degenerate case). *Let x^* be a minimizer with active constraints matrix $A_a \in \mathbb{R}^{n\times n}$, that is*

$$Ax^* \geq b, c = A_a^T y_a^*, y_a^* \geq 0.$$

If

$$c = A_a^T y_a^*$$

and

$$y_a^* > 0 \text{ (strict complementarity)},$$

then x^* is the unique minimizer, and it is also a vertex minimizer.

Proof. (1) From the equivalence between search direction and feasible directions [9.3.1], for a non-zero feasible direction p such that $A_a p > 0$(note that since A_a is nonsingular, $A_a p \neq 0$ if $p \neq 0$). Therefore,

$$c^T(x^* + \alpha p) = c^T x + \alpha y_a^* A p > c^T x^*.$$

(2) Because of the fact that if there exists minimizer, then there must exist vertex minimizer, the unique minimizer must be vertex minimizer. □

Theorem 9.3.3 (non-uniquenss of a non-degenerate vertex minimizer). *Let x^* be a **non-degenerate vertex** for the canonical linear programming problem. If*

$$c = A_a^T y_a^*, y_a^* \geq 0, \text{ and } [y_a^*]_i = 0 \text{ for at least one index } i,$$

then x^* is a vertex minimizer, but it is not unique.

Proof. Consider a search direction [Lemma 9.3.3] p at x^* such that

$$A_a p = e_i.$$

For $\alpha > 0$ sufficiently small, we have

$$c^T(x^* + \alpha p) = c^T x^* + \alpha c^T p = c^T x^* + \alpha p^T A_a^T y_a^a = c^T x^* + [y_a^*]_i = c^T x^*.$$

Since $x^* + \alpha p$ is feasible, therefore it is also a minimizer. Therefore, x^* is not unique. □

Definition 9.3.3 (working set, working matrix). *At the kth iterate x_k(a vertex). The **working set** $\mathcal{W}_k$ is a index set such that*

- *$\mathcal{W}_k$ contains **exactly** n indices, i.e., $\mathcal{W}_k = \{w_1, w_2, ..., w_n\}$.*
- *For every $j \in \mathcal{W}_k$, constraint j is active, i.e. $j \in \mathcal{A}(x_k)$.*

*The **working matrix** A_k is the $n \times n$ square **non-singular** matrix such that each row is active constraint matrix row.*

Remark 9.3.3 (geometry of working matrix and optimality)**.**

- Under non-degeneracy assumption, the working matrix A_k consists of the normals of hyperplanes intersecting at x_k.
- If $y_k \geq 0, y_k \in \mathbb{R}^n$, then x_k is the vertex minimizer(see the following lemma).

Lemma 9.3.2. *Let non-degeneracy assumption hold. Let x_k be the kth iterate (a vertex), and A_k be the working matrix. It follows that if*

$$y_k \geq 0, y_k \in \mathbb{R}^n$$

where y_k is the solution of

$$A_k y_k = c,$$

then x_k is the minimizer and (x_k, y_k) is the KKT point.

Proof. From the optimality condition [Theorem 9.2.1], since x_k is feasible, then $A_k x_k \geq b$ is satisfied. □

9.3.1.3 *Descent direction at a vertex*

Definition 9.3.4 (search direction at a vertex). *Let s be the index such that $[y_k]_s < 0$,, then p_k, solved from $A_k p_k = e_s$, is called **search direction**.*

Lemma 9.3.3 (properties of search direction). *Let x_k be the kth iterate, A_k be the working matrix, and $p_k, A_k p_k = e_s$ be the search direction, where s is the index $[y_k]_s < 0$. Then, given $\alpha > 0$, we have*

- *$x = x_k + \alpha p_k$, with α sufficiently small, will be still in the feasible region, and **inactivate** the sth constraint and maintain the rest $n-1$ active constraints active.*
- *moving in p_k direction will **decrease the objective value**; that is, $L(x) < L(x_k)$.*
- *If no such s exists, then x_k is optimal.*

Proof. (1)From

$$A_k x = A_k(x_k + \alpha p_k) = b + \alpha e_s \geq b$$

we know sth constraint is inactivated and the rest are still active. (2) $c^T x = c^T(x_k + \alpha p_k) = c^T x_k + \alpha c^T p_k = c^T x_k + \alpha (A_k^T y_k)^T p_k = c^T x_k + [y_k]_s < c^T x_k$. □

Note 9.3.1 (search direction is equivalent to the set of all feasible directions at a non-degenerate vertex)**.**

- Given a **vertex** x_k such that $Ax_k = b$,for $\alpha > 0$ sufficiently small and for all directions p such that $A_k p \geq 0$(note that $A_k \neq A$), the new point

$$x_0 + \alpha p$$

  will still be a feasible point($A_k(x_0 + \alpha p) \geq 0$ and $x_0 + \alpha p$ will not hit other inactive constraint for α sufficiently small) . Therefore, the set $\{p : A_k p \geq 0\}$ is a subset of the set of all possible feasible directions.
- The set of all possible feasible directions is also given by $\{p : A_k p \geq 0\}$(see Definition 8.4.5].

9.3.1.4 *Stepping along a descent direction*

Definition 9.3.5 (decreasing constraints in the search direction, blocking constraints). *The set of* ***decreasing constraints*** *$\mathcal{D}_k$ at the point x_k is given by*

$$\mathcal{D}_k \triangleq \{j : a_j^T p_k < 0\},$$

where a_j^T is the jth row of the constraint matrix.

Remark 9.3.4 (geometry of decreasing constraints). When we move along direction p_k, we might encounter hyperplanes(i.e. other constraints). The set of hyperplanes encountered are $\mathcal{D}_k$.

Moreover, if we do not encounter any hyperplanes, then **moving along direction p_k will decrease the objective function to -∞ and remain feasibility**.

Definition 9.3.6 (maximum feasible step). *The maximum feasible step α_k along a direction p_k is given by*

$$\alpha_k = \min_{j\in\mathcal{D}_k} \sigma_j,$$

where

$$\sigma_j = \frac{a_j^T x_k - b}{-a_j^T p_k}.$$

If $\mathcal{D}_k$ is an empty set, then move along direction p_k will decrease the objective function to -∞ and remain feasibility.

Remark 9.3.5 (geometric interpretation).

- The maximum feasible step is the maximum length we can move along direction p_k such that we can still maintain the feasibility of $x_{k+1} = x_k + \alpha_k p_k$.
- If we take the maximum feasible step along p_k, we are moving from one vertex to an adjacent vertex [Definition 9.3.2] in the feasible region.
- If we do not take a step $\alpha > \alpha_k$, then we lose the feasibility $x = x_k + \alpha_k p_k$ because we are violating the first constraint we encounter along p_k.

9.3.2 The simplex algorithm

Algorithm 14: The Simplex algorithm (non-degenerate system)

Input: Initial **vertex** x_0 with associated working set $\mathcal{W}_0$ and working matrix $\mathcal{A}_0$

Set $k = 0$

repeat

  Compute the unique Lagrange multiplier estimate y_k from $A_k^T y_k = c$.

  if $y_k \geq 0$ **then**

    return x_k as the minimizer

  end

  Choose an index s so that $[y_k]_s < 0$, and compute a search direction p_k such that

$$A_k p_k = e_s$$

  Compute the residual vector $r(x) = A_k x_k - b$

  Compute the set of decreasing constraints $\mathcal{D}_k\{j : a_j^T p_k < 0\}$ where a_j^T is the jth row of A.

  Compute the step lengths to the constraints as

$$\sigma_j = \begin{cases} \frac{r_j(x_k)}{-a_j^T p_k}, if\ j \in \mathcal{D}_k \\ +\infty, otherwise \end{cases}$$

  Set the maximum feasible step length as

$$\alpha_k = \min_{1 \leq j \leq m} \sigma_j.$$

  if $\alpha_k = +\infty$ **then**

    return because the objective function can reach -∞ on the feasible region

  end

  Choose t as the index of a blocking constraint satisfying $\sigma_t = \alpha_k$.

  Set $x_{k+1} = x_k + \alpha_k p_k$, $\mathcal{W}_{k+1} = \mathcal{W}_k - \{w_s\} + \{t\}$, and $A_{k+1} = A_k$ with the sth row replaced by the tth row of A.

  Set $k = k + 1$.

until *termination condition satisfied*;

Output: approximate minimizer x_k

9.4 Interior point method

9.4.1 Optimality condition

Definition 9.4.1 (standard form linear optimization). *A linear programming standard form is:*

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} c^T x \\ &\text{subject to } Ax = b \\ &\qquad x \geq 0 \end{aligned}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}, m \leq n, rank(A) = m$, *and* $b \in \mathbb{R}^m$.

Remark 9.4.1 (Why requires $m \leq n$ and $rank(A) = m$).

- Assume consistence of $Ax = b$. If A is not full row rank, then we can always eliminate dependence row.
- If $m \geq n, rank(A) = n$, then there is at most one feasible point, thus making the optimization trivial.

Lemma 9.4.1 (standard form linear programming, optimality condition, recap). *[3, p. 359] The optimality conditions are given by*

$$\begin{aligned} c &= A^T y + z, \\ Ax &= b, \\ x_i \cdot z_i &= 0, i = 1, ..., n \\ x &\geq 0, \\ z &\geq 0 \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, y \in \mathbb{R}^m, z \in \mathbb{R}^n, b \in \mathbb{R}^m$.

Proof. See Theorem 9.2.1. □

Note 9.4.1. Find the optimality condition is equivalent to finding a zero of a **nonlinear** equation $F : \mathbb{R}^{m+n+n} \to \mathbb{R}^{m+n+n}$, given by

$$F(x,y,z) = \begin{bmatrix} z + A^T y - c \\ Ax - b \\ XZ\mathbf{1} \end{bmatrix} \text{subject to } x \geq 0, z \geq 0$$

The Jacobian of F is given by

$$J = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix}$$

Remark 9.4.2. Usually a full affine step will violate the bound $(x,z) \geq 0$, so we perform a line search along the Newton direction and define the new iterate as

$$(x,y,z) = (x,y,z) + \alpha(\Delta x, \Delta y, \Delta z)$$

for some α such that bounds not being violated. Often $\alpha \ll 1$ to ensure $(x,z) \geq 0$. Therefore affine step usually cannot make large progress.

Lemma 9.4.2 (Jacobian matrix is non-singular if A is full row rank). *If A is full row rank and $x, z > 0$, the Jacobian matrix*

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix}$$

is nonsingular.

Proof. We want to show that

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0,$$

only has solution $\Delta x = \Delta y = \Delta z = 0$. From the 2nd row, we know that $\Delta x \in \mathcal{N}(A)$, and from the first row we know that $\Delta z \in \mathcal{R}(A^T)$. Therefore, $\Delta x^T \Delta z = 0$. From the third row, we have

$$Z\Delta x + X\Delta z = 0 \implies \Delta x = -Z^{-1}X\Delta z \implies 0 = \Delta z^T \Delta x = -\Delta z^T Z^{-1} X \Delta z.$$

Because $Z^{-1}X > 0$, we have $\Delta z = 0$. Then, $Z\Delta x + 0 = 0 \implies \Delta x = 0$. From the first row $A^T \Delta y = 0 \implies \Delta y = 0$ since A^T is full column rank. □

9.4.2 Newton step and perturbed system

Definition 9.4.2 (perturbed optimality conditions). *The perturbed optimality conditions with perturbation parameter $\tau > 0$ are given by*

$$\begin{aligned} c &= A^T y + z, \\ Ax &= b, \\ x_i \cdot z_i &= \tau, i = 1, ..., n \\ x &\geq 0, \\ z &\geq 0 \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, y \in \mathbb{R}^m, z \in \mathbb{R}^n, b \in \mathbb{R}^m$.

Remark 9.4.3 (existence and uniqueness of perturbed solutions).

Remark 9.4.4 (log-barrier interpretation). [3, p. 397] The perturbed optimality condition is equivalent to solve

$$\min_x c^T x - \tau \sum_{i=1}^n \ln x_i, \textit{ subject to } Ax = b, x \geq 0.$$

As $\tau \to 0$, we are approach the original standard linear optimization.

Definition 9.4.3 (Primal-dual feasible region). *The primal-dual* ***feasible set*** *$\mathcal{F}$ and* ***strictly feasible set*** *$\mathcal{F}^0$ are defined as*

$$\begin{aligned} \mathcal{F} &= \{(x, y, z) : Ax = b, c = A^T y + z, (x, z) \geq 0\} \\ \mathcal{F}^0 &= \{(x, y, z) : Ax = b, c = A^T y + z, (x, z) > 0\} \end{aligned}$$

Remark 9.4.5 (Primal-dual feasible region are almost the optimality condition).

- Note that the **Primal-dual feasible region** is much more restrict than the **primal feasible region**(i.e., $Ax = b, x \geq 0$).
- The Primal-dual feasible region satisfies almost all the optimality condition [Theorem 9.2.1] except for the complementary condition.

Definition 9.4.4 (central path in the primal-dual feasible region). *The central path $\mathcal{C}$ associated with the standard LP is defined as*

$$\mathcal{C} = \{(x_\tau, y_\tau, z_\tau] : \tau > 0\},$$

where (x_τ, y_τ, z_τ) satisfies

$$c = A^T y + z, Ax = b, x_i \cdot z_i = \tau, x_i \geq 0, z_i \geq 0, i = 1, ..., n,$$

*for some value $\tau > 0$. **The central path is always in primal-dual feasible region and bounded away from the boundary of the feasible set.***

Definition 9.4.5 (neighborhood of the central path). *The most common neighborhood are*

$$N_2(\theta) = \{(x, y, z) \in \mathcal{F}^0 : \left\|XZe - \mu e\right\|_2 \leq \theta\mu\}(two - norm\ neighborhood)$$
$$N_{-\infty}(\gamma) = \{(x, y, z) \in \mathcal{F}^0 : x_i z_i \geq \gamma\mu, \forall i = 1, ..., n\}(wide\ neighborhood)$$

for some constants $\theta \in [0, 1)$, and $\gamma \in (0, 1)$. Typical values used in practice are $\theta = 0.5, \gamma = 10^{-3}$.

Remark 9.4.6. Those Central neighborhoods are all subsets of strictly feasible set.

Definition 9.4.6 (Newton system for perturbed system). *The Newton system for the perturbed optimality condition is given by*

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} A^T y + z - c \\ Ax - b \\ XZ\mathbf{1} - \tau\mathbf{1} \end{bmatrix}$$

Remark 9.4.7 (Centering step vs. affine step). [3, p. 398]

- When $\tau = 1$, the new value of (x, y, z) is **getting closer to the central path, which is always in feasible region and bounded away from the boundary of the feasible set.**

- Centering directions are usually biased strongly toward the interior of the non-negative orthant and make little progress in reducing the duality measure $\mu = x^T z$.
- However, make a centering step will probably set the scene for a substantial reduction in μ in the next iteration.
- When $\tau = 0$, we are taking an affine step, which usually lead to large reduction of μ but might hit the boundary $(x, z \geq 0)$. Usually, if $\tau > 0$, it is possible to take a longer step α along the direction $(\Delta x, \Delta y, \Delta z)$ before violating the bounds $(x, z) \geq 0$.

9.4.3 Algorithms

Algorithm 15: Primal-dual long-step path-following algorithm

Input: Initial estimate $(x_0, y_0, z_0) \in N_{-\infty}(\gamma)$. Choose parameter $\gamma \in (0,1)$ and $0 < \sigma_{min} < \sigma$

1 Set $k = 0$

2 **repeat**

3 Compute the duality measure $\mu_k \triangleq (x_k^T z_k)/n$. Choose $\sigma_k \in [0,1]$ and compute a trial step $(\Delta x, \Delta y, \Delta z)$ satisfying

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} A^T y_k + z_k - c \\ A x_k - b \\ X_k Z_k \mathbf{1} - \sigma_k \mu_k \mathbf{1} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ X_k Z_k \mathbf{1} - \sigma_k \mu_k \mathbf{1} \end{bmatrix}$$

4 Choose α_k as the largest value in $(0,1]$ such that

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k, y_k, z_k) + \alpha_k (\Delta x, \Delta y, \Delta z).$$

such that $(x_{k+1}, y_{k+1}, z_{k+1}) \in N_{-\infty}(\gamma) = \{(x, y, z) \in \mathcal{F}^0 : x_i z_i \geq \gamma\mu, \forall i = 1, ..., n\}$.

5 Set $k = k + 1$.

6 **until** *termination condition satisfied*;

Output: approximate minimizer x_k

Remark 9.4.8 (interpretation).

- Duality measure $\mu \triangleq x^T z / n$ is not the duality gap, it simply measure how far we are away from optimality $\mu_{opt} = 0$.
- We can terminate the algorithm when μ is sufficiently small.

Remark 9.4.9 (convergence). [3, p. 406]

- It can be showed that the dual measure μ_k is decreasing "sufficiently" for each step therefore as $k \to \infty$, $\mu_k \to 0$ and (x_k, y_k, z_k) converge to the optimal solution.
- The convergence speed is linear.

Remark 9.4.10 (other algorithms). See [3, p. 406] for more algorithms.

9.5 Notes on bibliography

The major references are [1] [4] [2].

For interior point method, see [3].

BIBLIOGRAPHY

1. Robinson, D. *Nonlinear optimization II lecture notes* (Johns Hopkins University, 2015).
2. Bertsimas, D. & Tsitsiklis, J. N. *Introduction to linear optimization* (Athena Scientific Belmont, MA, 1997).
3. Nocedal, J. & Wright, S. *Numerical optimization* (Springer Science & Business Media, 2006).
4. Luenberger, D. & Ye, Y. *Linear and Nonlinear Programming* ISBN: 9783319188423 (Springer International Publishing, 2015).

10

CONVEX ANALYSIS AND CONVEX OPTIMIZATION

10.1 Affine sets

10.1.1 Basic concepts

Definition 10.1.1 (affine sets). *A set* $\subset \mathbb{R}^d$ *is affine set if for all* $x, y \in X$ *and* $\lambda, \gamma \in \mathbb{R}, \lambda + \gamma = 1$, *then* $\lambda x + \gamma y \in X$

Lemma 10.1.1. *Arbitrary intersection of affine sets is an affine set.*

Proof. Let $x, y \in X \cap Y$, then $\lambda x + \gamma y \in X \cap Y$. □

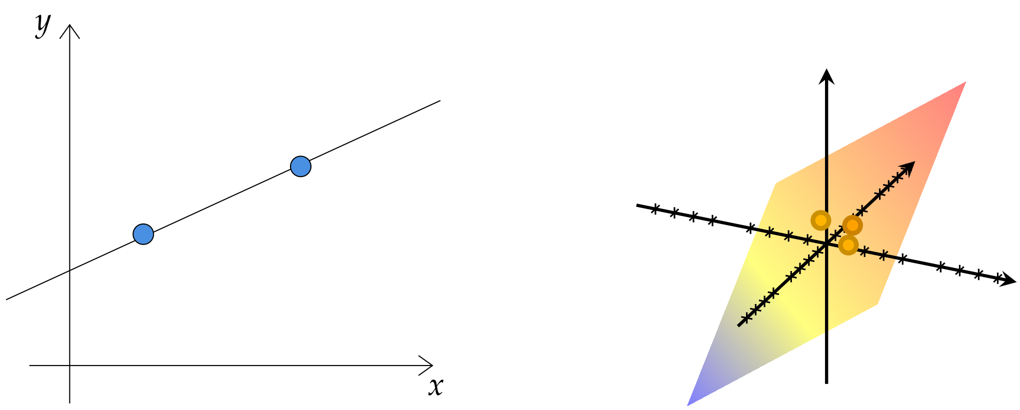
Definition 10.1.2 (affine combination). *Given* $x, y \in \mathbb{R}^d$, *the affine combination* z *of* x, y *refers to*

$$z = ax + by, a + b = 1$$

Definition 10.1.3 (affine hull). *The affine hull of* X *is defined as*

$$aff(X) = \{\lambda_1 x_1 + ... + \lambda_t x_t : x_1, ..., x_t \in X, \sum_i \lambda_i = 1\}$$

That is, the affine hull the the set of all affine combinations of points in X.



(a) The affine hull of two points in a plane is a line passing through them.

(b) The affine hull of three co-plane points is a plane containing them them.

Figure 10.1.1: Example 2D affine hull and 3D affine hull.

Lemma 10.1.2 (characterization via linear equations). *[1, p. 5]*

- *The solution to the linear equation $Ax = 0$ form a linear subspace X. Moreover, every linear subspace set can be represented as*
$$\{x : Ax = 0\}$$
- *The solution to the linear equation $Ax = b$ form an affine subspace X. Moreover, every affine set can be represented as*
$$\{x : Ax = b\}$$

Proof. (1) (a) nullspace is subspace; (b) consider the orthogonal complement $X^\perp$, and let basis of $X^\perp$ be the rows of A. (2) (a) Let x_1, x_2 be such that $Ax_1 = b, Ax_2 = b$, then

$$A(ax_1 + (1-a)x_2) = ab + (1-a)b = b$$

therefore $ax_1 + ax_2$ is also the solution. Thus the solution form an affine subspace. (b) Let L be the associated linear subspace, then consider the orthogonal complement $L^\perp$, and let basis of $L^\perp$ be the rows of A. Since $X = L + a$, for some $a \in \mathbb{R}^d$, then $X = \{x : A(x-a) = 0\} = \{x : Ax = Aa\}$. □

Definition 10.1.4 (parallel relation in affine sets). *An affine set M is said to be* ***parallel*** *to an affine set L if there exist some a such that*

$$M = L + a$$

Lemma 10.1.3 (translational invariance in affine subspace). *If $X \subset \mathbb{R}^d$ is an affine subspace, then for every $x_0 \in \mathbb{R}^d$, $X - x_0$ is still an affine subspace. That is, affine subspace is translational invariant, and shifted affine subspaces are parallel to each other.*

Proof. Let $x, y \in X$, then $x - x_0, y - x_0 \in X - x_0$, and then we can verify

$$a(x - x_0) + b(x - x_0), a + b = 1$$

also belongs to $X - x_0$. □

Lemma 10.1.4 (affine subspace to linear subspace). *[1, p. 4] If an affine subspace X contains 0, then X is also an linear space.*

Proof. (1) closedness under scalar multiplication: Let $x \in X$, then

$$\lambda x + (1-\lambda)0 = \lambda x \in X, \forall \lambda \in \mathbb{F}$$

(2) closedness under addition: Let $x, y \in X$, then $x + y = 2(\frac{1}{2}(x+y)) \in X$, since $\frac{1}{2}(x+y) \in X$ and X is closed under scalar multiplication. Other properties can be verified easily. □

Lemma 10.1.5 (conversion from affine subspace to linear subspace). *If $X \subset \mathbb{R}^d$ is an affine subspace, then for every $x_0 \in X$, $X - x_0$ is a linear space. Moreover,*

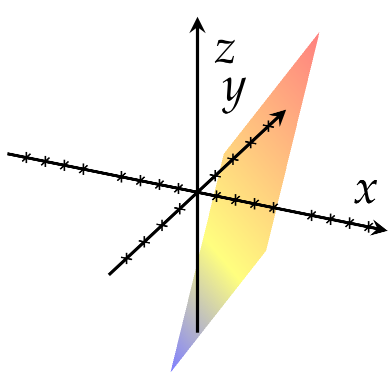
$$X - x_1 = X - x_2, \forall x_1 \neq x_2, x_1, x_2 \in X$$

Proof. (1) $X - x_0$ is an affine subspace containing 0. (2) Let $L_1 = X - x_1$ and $L_2 = X - x_2$, then $L_1 = L_2 + a, a = x_1 - x_2$. Then since L_1 contains 0 then $-a \in L_2$, further implying $a \in L_2$, because $L_2 = a + L_2, a \in L_2$ due to closedness of vector space, therefore

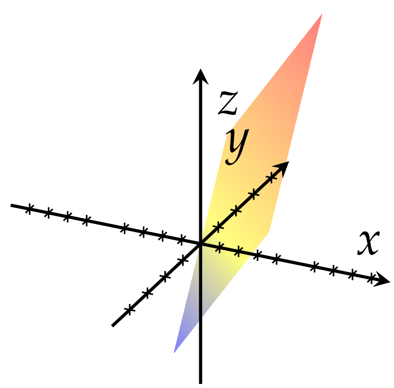
$$L_1 = L_2$$

□

Corollary 10.1.0.1 (associated unique linear subspace). *Every non-empty affine set M is parallel to a unique linear subspace L, which is given as $L = M - x, x \in M$.*



(a) An affine 2D surface that does not contain the origin.



(b) An affine 2D surface that contains the origin; thus it is also a subspace.

Figure 10.1.2: Affine subspace and linear subspace.

10.1.2 Affine independence and dimensions

Definition 10.1.5 (dimension of an affine set). *[1, p. 4] The dimension of an affine set is defined as the dimension of its parallel linear subspace.*

Definition 10.1.6 (affine independence). *A set X is affinely independent if there is no $x \in X$ such that $x \in aff(X - \{x\})$. That is, for every element in X, it can not be written as affine combination of the others.*

Definition 10.1.7 (affine independence,alternative). *[1, p. 7] A set X of $m+1$ points $b_0, b_1, ..., b_m$ are affinely independent if $aff(X)$ has dimension of m; that is the linear subspace*

$$X - b_i, i \in \{0, 1, ..., m\}$$

has dimension of m; or equivalently, the set

$$b_1 - b_0, b_2 - b_0, ..., b_m - b_0$$

are linearly independent.

Lemma 10.1.6 (algebraic criterion for affine independence). *[2, p. 23] The elements $x_0, ..., x_m$ in $\mathbb{R}^n, m \geq 1$, are affinely independent if and only if*

$$\sum_{i=1}^{m} \lambda_i v_i = 0, \sum_{i=0}^{m} \lambda_i = 0 \Rightarrow \lambda_i = 0, \forall i = 0, ..., m$$

Proof. (1) Suppose

$$\sum_{i=0}^{m} \lambda_i v_i = 0, \sum_{i=0}^{m} \lambda_i = 0 \Rightarrow \lambda_i = 0, \forall i = 0, ..., m$$

We can rewrite as

$$\sum_{i=1}^{m} \lambda_i (v_i - v_0) = 0 \Rightarrow \lambda_i = 0, \forall i = 1, ..., m$$

which implies $v_i - v_0, i = 1, ..., m$ are linearly independent. Therefore $v_0, ..., v_m$ are affinely independent. (2) Suppose affinely independent. WLOG, suppose $\lambda_0 \neq 0$, we have

$$\lambda_0 v_0 = -\sum_{i=1}^{m} \lambda_i v_i$$

add both sides with $\sum_{i=1}^{m} \lambda_i v_0$, we have

$$0 = \sum_{i=1}^{m} \lambda_i (v_i - v_0)$$

implies $\lambda_i = 0, i = 1, ..., m \Rightarrow \lambda_0 = 0$, contradicts $\lambda_0 \neq 0$. $\square$

Remark 10.1.1 (linear space analog). A set X is linearly independent if every $x \in X$ cannot be written as the linear combination of the others. Or there is no $x \in X$ such that $x \in span(X - \{x\})$

Theorem 10.1.1 (Barycentric coordinate system). *Let $b_0, b_1, ..., b_m$ be affinely independent, let $M = aff(b_0, ..., b_m)$. Then for any $x \in M$, we have a* ***unique affine combination*** *representation for x, given as*

$$x = \sum_{i=0}^{m} a_i b_i, \sum_{i=0}^{m} a_i = 1, a_i \in \mathbb{R}$$

We can view this as the Barycentric coordinate system.

Proof. Because $b_0, b_1, ..., b_m$ be affinely independent, then the set

$$b_1 - b_0, ..., b_m - b_0$$

are linearly independent. Because $x \in M$, then x has representation of

$$x = \sum_{i=0}^{m} a_i b_i, \sum_{i=0}^{m} a_i = 1$$

To show uniqueness, we subtract out b_0 on both side, we have

$$x - b_0 = \sum_{i=1}^{m} a_i (b_i - b_0)$$

Suppose we also have

$$x - b_0 = \sum_{i=1}^{m} c_i (b_i - b_0)$$

Then we have

$$\sum_{i=1}^{m} (c_i - a_i)(b_i - b_0) = 0$$

implies $c_i = a_i, i = 1, 2, ..., m$ due to linear independence of $b_1 - b_0, ..., b_m - b_0$. $\square$

Theorem 10.1.2. *Let $X \subset \mathbb{R}^d$. Then the following are equivalent:*

- *X is an affinely independent*
- *For every $x \in X$, the set $\{v - x : v \in X - \{x\}\}$ is linearly independent.*
- *There exists $x \in X$ such that the set $\{v - x : v \in X - \{x\}\}$ is linearly independent.*
- *The set of vectors $\{(x, 1) \in \mathbb{R}^{d+1}, x \in X\}$ is linearly independent.*
- *X is finite set with vectors $x_1, ..., x_m$ such that*

$$\lambda_1 x_1 + ... + \lambda_m x_m = 0, \lambda_1 + ... + \lambda_m = 0$$

  implies $\lambda_1 = ... = \lambda_m = 0$

Theorem 10.1.3. *Let $X \subset \mathbb{R}^d$. The following are equivalent.*

- *X is an affine subspace*
- *For every $x \in X$, the set $X - x$ is a linear subspace of dimension $0 \leq m \leq d$*
- *There exist affinely independent vectors $x_1, ..., x_{m+1}$ for some $0 \leq m \leq d$ such that every $x \in X$ can be written as $x = \sum_i \lambda_i x_i, \sum_i \lambda_i = 1$*
- *There exists a matrix $A \in \mathbb{R}^{(d-m)\times d}$ with full row rank and a vector $b \in \mathbb{R}^m$ for some $0 \leq m \leq d$ such that*

$$X = \{x \in \mathbb{R}^d : Ax = b\}$$

10.2 Convex sets and properties

10.2.1 Concepts of convex sets

Definition 10.2.1 (convex set, strictly convex set).

- *A set S is said to be a convex set if and only if for any two points $x, y \in S$, we have*
$$\alpha x + (1-\alpha)y \in S, \forall \alpha \in [0,1].$$
- *A set S is said to be a strictly convex set if and only if for any two points $x, y \in S$, we have*
$$\alpha x + (1-\alpha)y \in intS, \forall \alpha \in (0,1),$$
where $intS$ is the interior of S.

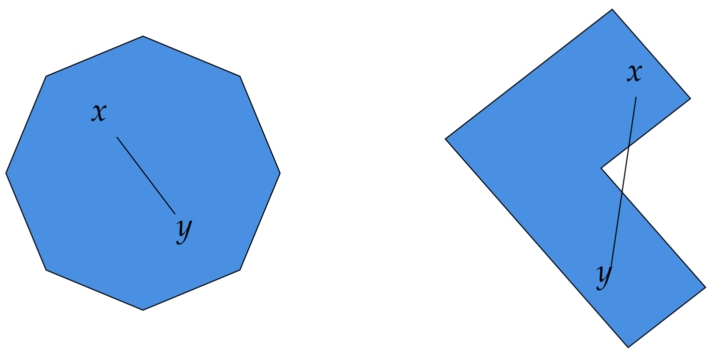


Figure 10.2.1: (left) A convex set. (right) A non-convex set.

Example 10.2.1.

- A convex set can be either open or closed. For example, (a,b) and $[a,b]$ are both convex sets.
- Let $a \in \mathbb{R}^d, \delta \in \mathbb{R}$, then the set $H = \{x \in \mathbb{R}^d : \langle a, x \rangle = \delta\}$ is called hyperplane. The set $H^+ = \{x \in \mathbb{R}^d : \langle a, x \rangle \geq \delta\}$ and $H^- = \{x \in \mathbb{R}^d : \langle a, x \rangle \leq \delta\}$ are halfspaces. Hyperplanes and halfspaces are all convex sets.

Definition 10.2.2 (dimensionality of a convex set). *Let X be a convex set, then $dim(X)$ equals to the maximum number of affinely independent points in X.*

Lemma 10.2.1 (preservation of convexity).

- *The intersection of **arbitrary** collection of convex sets $\cap_{i\in I}C_i$ is a convex set.*
- *The image of any linear function f defined on a convex set X is convex.*
- *If X is convex, then the set $aX = \{ax : x \in X\}, a \in \mathbb{R}$ is also convex. Particularly, $-X$(the symmetric set with respect to origin) is convex.*
- *If X, Y are convex, then $X + Y$ and $X - Y = X + (-Y)$ is also convex.*

Proof. (1) Let $x, y \in \cap_{i\in I}C_i$, then its convex optimization belongs to any convex set C_i, and therefore in $\cap_{i\in I}C_i$.(2)$af(x) + (1-a)f(y) = f(ax + (1-a)y) \in Im(f)$. (3)(4) directly follow from definition,(4) will use (3). □

Remark 10.2.1 (Empty set is convex). Note that intersection might result in empty set. Even so, it does not violate above lemma, since empty set is also convex.

Lemma 10.2.2. *[3, p. 44] A convex combination of a finite number of elements of a convex set X also belongs to that set.*

Proof. Use induction to prove, starting from n=2. Suppose $n = k$ holds, then for $n = k+1$, we have

$$\lambda_0 a_0 + \lambda_1 a_1 + ... + \lambda_k a_k = \lambda_0 a_0 + (1-\lambda_0)(\frac{\lambda_1}{1-\lambda_0}a_1 + ... + \frac{\lambda_k}{1-\lambda_0}a_k) \in X, \sum_i \lambda_i = 1$$

since $(\frac{\lambda_1}{1-\lambda_0}a_1 + ... + \frac{\lambda_k}{1-\lambda_0}a_k) \in X$ by assumption. □

Theorem 10.2.1 (convex hull as the smallest containing affine set). *The convex hull of X, denoted as $conv(X)$,given as*

$$conv(X) = \{\lambda_1 x_1 + ... + \lambda_t x_t : x_1, ..., x_t \in X, \lambda_1, ..., \lambda_t \geq 0, \sum_i \lambda_i = 1\}$$

is the smallest convex hull containing X. Moreover, a set X is convex if $X = conv(X)$.

Proof. It is easy to see $X \subseteq conv(X)$. To prove it is the smallest: let C be any other convex set that containing X, then we have $conv(X) \subseteq C$ due to above lemma. □

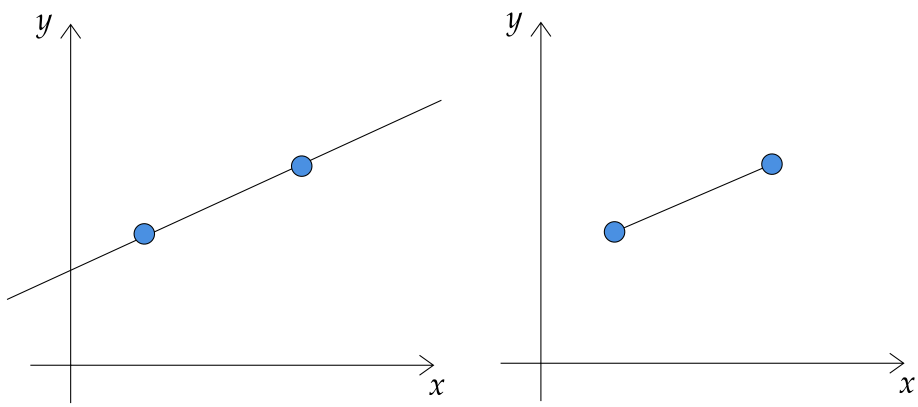


Figure 10.2.2: (left) The affine hull of two points in a plane is a line passing through them. (right) The convex hull of two points in a plane is a line segment containing them them.

10.2.2 Projection theorems

Definition 10.2.3 (projection onto a convex set). *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a nonempty closed convex set. The projection of a given vector $y \in \mathbb{R}^n$ onto $\mathcal{X}$ is*

$$Proj_{\mathcal{X}}(y) = \arg\min_{x \in \mathcal{X}} \|x - y\|_2 .$$

Theorem 10.2.2 (projection theorem for convex set). *[4, p. 19] Let $\mathcal{X}$ be a closed convex set in $\mathbb{R}^n$, let $\|.\|$ be the Euclidean norm. Then, we have*

- *For every $x \in \mathbb{R}^n$, the following optimalization problem*

$$\min_{y \in X} \|y - x\|^2$$

  has an unique global minimum solution x^*.
- *(obtuse angle theorem) $x^* = Proj_{\mathcal{X}}(y), y \in \mathbb{R}^n$ is the unique global minimum if and only if*

$$(y - x^*)^T (x - x^*) \leq 0, \forall x \in \mathcal{X}$$

- *The projection mapping $Proj_{\mathcal{X}}(y) : \mathbb{R}^n \to \mathcal{X}$ is continuous and non-expansive mapping, i.e.*

$$\|Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2)\| \leq \|y_1 - y_2\| .$$

Proof. (1) For every $x \in \mathcal{X}$, it is easy to see that the function $f(y) = \|y - x\|^2$ is coercive in the set $\mathcal{X}$. Therefore, from Theorem 3.1.7, we have at least one global minimizer in $\mathcal{X}$.

This minimizer will be unique [Lemma 10.5.2] since $f(y) = \|y - x\|^2$ is strictly convex(The Hessian is positive definite matrix).
(2) Use the first order necessary and sufficient condition for $f(x)$ [section 10.5] gives

$$\nabla f^T(x - x^*) \geq 0, \forall x \in \mathcal{X} \Leftrightarrow (x - x^*)^T(x^* - y) \geq 0, \forall x \in \mathcal{X}$$

(3) From (2), we have

$$\begin{aligned} &(y_1 - Proj_{\mathcal{X}}(y_1))^T(x - Proj_{\mathcal{X}}(y_1)) \leq 0, \forall x \in \mathcal{X} \\ \implies &(y_1 - Proj_{\mathcal{X}}(y_1))^T(Proj_{\mathcal{X}}(y_2) - Proj_{\mathcal{X}}(y_1)) \leq 0, \forall x \in \mathcal{X} \\ &(y_2 - Proj_{\mathcal{X}}(y_2))^T(Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2)) \leq 0, \forall x \in \mathcal{X} \end{aligned}$$

Add together, we have

$$\begin{aligned} &-(Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2))^T(y_1 - y_2 - (Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2))) \leq 0 \\ &\|Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2)\|^2 \leq (Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2))^T(y_1 - y_2) \\ &\qquad \leq \|Proj_{\mathcal{X}}(y_1) - Proj_{\mathcal{X}}(y_2)\| \|y_1 - y_2\| \end{aligned}$$

where we use the Cauchy inequality.

Nonexpansiveness directly implies continuity: as $y_1 \to y_2$, $Proj_{\mathcal{X}}(y_1) \to Proj_{\mathcal{X}}(y_2)$. □

10.2.3 Separation theorems

10.2.3.1 *Separating hyperplane theorem*

Theorem 10.2.3 (separating hyperplane theorem). *[3, p. 170] Let $C \subset \mathbb{R}^d$ be a closed convex set and let $x \in \mathbb{R}^d, \notin C$. Then there exists a halfspace that contain C and does not contain x. More precisely, there exists $a \in \mathbb{R}^d, \delta \in \mathbb{R}$ such that $\langle a, y \rangle \leq \delta$ for all $y \in C$ and $\langle a, x \rangle > \delta$; that is*

$$\langle a, x \rangle > \langle a, y \rangle, \forall y \in C;$$

or equivalently,

$$\langle a, x \rangle > \sup_{y \in C} \langle a, y \rangle.$$

Proof. (1) If C is empty, then any halfspace that does not contain x will suffice. (2) If C is empty, consider a closed ball $B(x, r)$,where $r = \|x - \bar{x}\|, \bar{x} \in C$, then the set $B(x, r) \cap C$ is a closed and compact set, and therefore there exists x^* as the minimizer of problem $\min_{y \in C \cap B} \|y - x\|$. Let $a = x - x^*$, then from obtuse angle theorem(projection theorem)Theorem 10.2.2 for convex set, we have

$$\langle y - x^*, x - x^* \rangle \leq 0$$

which implies

$$\langle y - x^*, a \rangle \leq 0 \Leftrightarrow \langle a, y \rangle \leq \langle a, x^* \rangle < \langle a, a + x^* \rangle = \langle a, x \rangle$$

□

Corollary 10.2.3.1 (separating hyperplane theorem for convex bodies). *Let C_1 and C_2 be two convex subsets in $\mathbb{R}^n$. If C_1 and C_2 are disjoint, then there exists $a \in \mathbb{R}^n$ such that*

$$\langle a, x_1 \rangle \leq \langle a, x_2 \rangle, \forall x_1 \in C_1, x_2 \in C_2$$

Proof. Consider the convex set $C = C_1 + (-C_2)$ (see Lemma 10.2.1]. Since C does not contain origin, from Theorem 10.2.3, there exists $a \in \mathbb{R}^m, \delta \in \mathbb{R}^m$ such that

$$\langle a, x \rangle \leq \langle a, 0 \rangle = 0 \Rightarrow \langle a, x_1 - x_2 \rangle \leq 0, \forall x_1 \in C_1, x_2 \in C_2$$

□

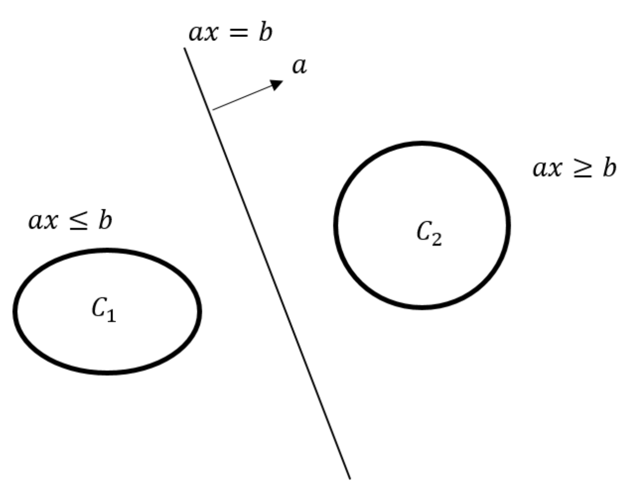


Figure 10.2.3: An illustration of separating hyperplane theorem for two convex bodies.

Corollary 10.2.3.2. *For every closed convex set $X \subseteq \mathbb{R}^d$ there exists a family of tuples $(a^i, \delta^i), a^i \in \mathbb{R}^d, \delta^i \in \mathbb{R}, i \in I$(where I may be an uncountable index set) such that $X =$*

$\cap_{i\in I} H^-(a^i, \delta^i)$. *In otherwise, every closed convex set can written as the intersection of some family of halfspaces.*

Corollary 10.2.3.3. *Every closed convex set C can e written as the intersection of some family of halfspaces.*

Proof. Consider all the points x_i outside C, then based on separating hyperplane theorem, there exists an halfspaces $H^-(a_i, \delta_i)$, such that $C \subseteq H^-(a_i, \delta_i)$. The intersection of all such halfspaces will be C since every point outside C will not be included in the intersection. □

Definition 10.2.4 (supporting hyperplane). *Given a set $C \subseteq \mathbb{R}^n$. If a vector x_0 belongs to $cl(C)$, a hyperplane with parameter $a \in \mathbb{R}^n, \delta \in R$ such that*

$$\langle a, x \rangle \leq \delta, \langle a, x_0 \rangle = \delta$$

is called a supporting hyperpalne for C.

Theorem 10.2.4 (supporting hyperplane theorem). *[5, p. 67] Let $C \subseteq \mathbb{R}^d$ be a convex set and let $x \in bd(C)$ (the boundary of C). Then there exists $a \in \mathbb{R}^d, \delta \in \mathbb{R}$ such that*

- *$\langle a, y \rangle \leq \delta$ for all $y \in C$; that is, $C \subseteq H^-(a, \delta)$*
- *$\langle a, x \rangle = \delta$*

The hyperplane $\{y \in \mathbb{R}^d : \langle a, y \rangle = \delta\}$ is called a supporting hyperplane for C at x.

Proof. See reference. □

10.2.3.2 *Farka's lemma*

Theorem 10.2.5 (Farkas's lemma, analog in linear equation theory). *Let $A \in \mathbb{R}^{m\times d}$ and $b \in \mathbb{R}^m$. Exactly one of the following is true:*

- *$Ax = b$ has a solution*
- *There exists $u \in \mathbb{R}^m$ such that $A^T u = 0$ and $u^T b \neq 0$*

Proof. (1) If $Ax = b$ has a solution, then $b \in \mathcal{R}(A)$. Suppose there exists a u such that $A^T u = 0$, which implies $u \in \mathcal{N}(A^T)$. Since $b \in \mathcal{R}(A), u \in \mathcal{N}(A^T), \mathcal{R}(A) \perp \mathcal{N}(A^T)$, it is impossible to have $u^T b \neq 0$;
(2) If $A^T u = 0$ has a solution u and $u^T b \neq 0$. Then b has nonzero component in $\mathcal{N}(A^T)$, and therefore b cannot lie in the $\mathcal{R}(A)$. □

Theorem 10.2.6 (Farkas' lemma). *Let $A \in \mathbb{R}^{m\times d}$ and $b \in \mathbb{R}^m, b \neq 0$. Exactly one of the following is true:*

- *$Ax = b, x \geq 0$ has a solution.*
- *There exists $u \in \mathbb{R}^m$ such that $A^T u \leq 0$ and $u^T b > 0$.*

  Or equivalently, exactly one of the following sets must be empty

- *$\{x | Ax = b, x \geq 0\}$.*
- *$\{u | A^T y \leq 0, b^T y > 0\}$.*

Proof. (1) Suppose $Ax = b, x \geq 0$ holds, then we suppose that there exists $u \in \mathbb{R}^m$ such that $A^T u \leq 0$. Multiply x^T, we have $(Ax)^T u \leq 0 \Rightarrow b^T u \leq 0$, which contradicts $u^T b > 0$. Therefore, when the first case holds, the second case cannot hold at the same time. (2) We can view each column a_i of A is a point in $\mathbb{R}^m$, then the set $C = \{y = Ax, x \geq 0\}$ form a cone. $Ax = b, x \geq 0$ has a solution can be interpreted as b is lying in the cone. Suppose b lying outside the cone, then use Theorem 10.2.3 we know that there exists $u \in \mathbb{R}^m, \delta \in \mathbb{R}$ such that

$$\langle y, u \rangle \leq \delta, \forall y \in C, \langle u, b \rangle > \delta$$

Also, 0 is in the cone, we have $\delta \geq 0$, therefore $\langle u, b \rangle > 0$. To show $\langle a_i, u \rangle \leq 0, \forall i$, suppose there exist some $\langle a_i, u \rangle > 0$, then there exists some scalar $\lambda > 0$ (we can always choose λ large enough) such that $\lambda \langle a_i, u \rangle > \delta$, then $\langle \lambda a_i, u \rangle > \delta$, contradicting the fact that every point in C satisfying $\langle \lambda a_i, u \rangle \leq \delta$. Therefore, $\langle a_i, u \rangle \leq 0, \forall i$ $\square$

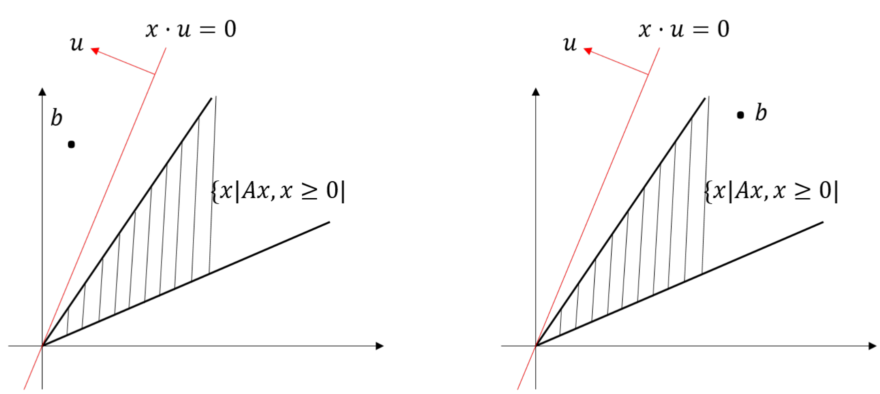


Figure 10.2.4: An illustration of Farkas' lemma. (left) When b lies outside the cone (that is, $Ax = b, x \geq 0$ has no solution), there exists a hyperplane, characterized by normal vector y, seperating b and the cone. (right) When b lies inside the cone (that is, $Ax = b, x \geq 0$ has a solution), there does not exist a hyperplane, characterized by normal vector y, seperating b and the cone.

Corollary 10.2.6.1 (Farkas' lemma variant). *Let $A \in \mathbb{R}^{m\times d}$ and $b \in \mathbb{R}^m, b \neq 0$. Exactly one of the following is true:*

- *$Ax = b, x > 0$ has a solution.*
- *There exists $u \in \mathbb{R}^m$ such that $A^T u \leq 0$ and $u^T b > 0$ (or $A^T u < 0$ and $u^T b \geq 0$).*

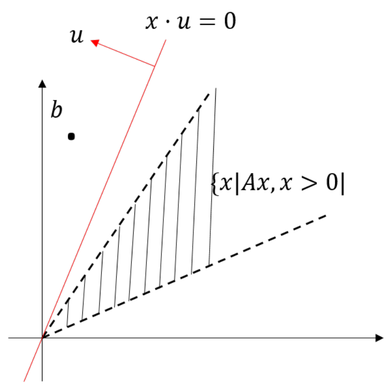


Figure 10.2.5: An illustration of Farkas' lemma variant where the cone is open set. (left) When b lies outside the cone (that is, $Ax = b, x > 0$ has no solution), there exists a hyperplane, characterized by normal vector y, seperating b and the cone. (right) When b lies inside the cone (that is, $Ax = b, x > 0$ has a solution), there does not exist a hyperplane, characterized by normal vector y, seperating b and the cone.

Remark 10.2.2 (financial application)**.** Farkas' lemma in no-arbitrage pricing can be found in [3, p. 168].

10.3 Convex functions

10.3.1 Basic concepts

Definition 10.3.1 (convex function, concave function).

- *A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex function** is for $x_1, x_2 \in \mathbb{R}^n, \alpha \in [0,1]$, then*

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2).$$

- *A function f is **concave** if $-f$ is convex.*
- *A function $f : \mathbb{R}^n \to \mathbb{R}$ is **strictly convex function** is for $x_1, x_2 \in \mathbb{R}^n, x_1 \neq x_2, \alpha \in (0,1)$, then*

$$f(\alpha x_1 + (1-\alpha)x_2) > \alpha f(x_1) + (1-\alpha) f(x_2).$$

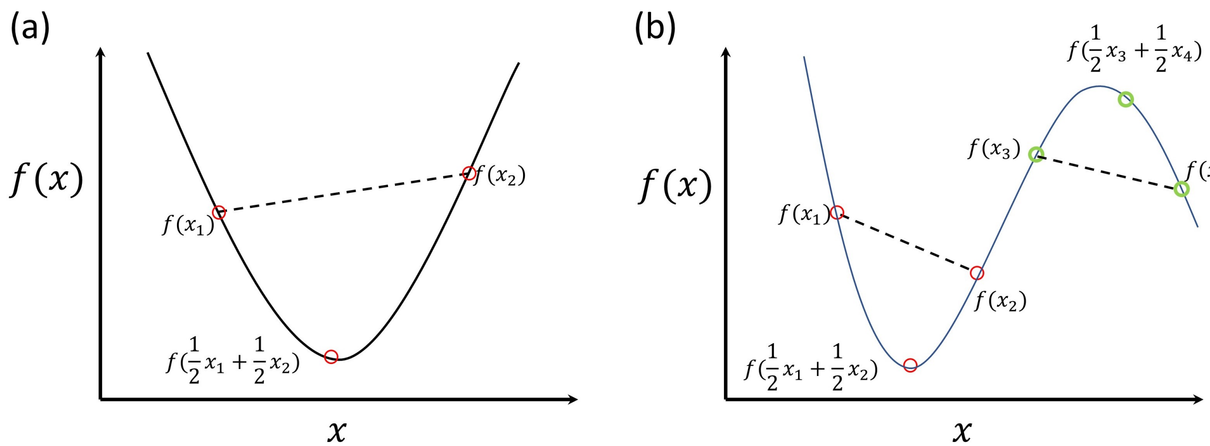


Figure 10.3.1: Demonstration of convex functions. (a) A convex function satisfying $f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$. (b) A non-convex function where the green points does not satisfy the relation.

Note 10.3.1 (the domain needs to convex). **Note that the domain C here also has to be convex.** We emphasize that the domain have to convex to avoid the situation that when x_1 and x_2 are in S but their convex combination is not in S.

Example 10.3.1 (Examples of convex and concave functions).

- Affine functions $Ax + b$ are both convex and concave.
- Power function x^a is convex for $a \geq 1$. concave if $a \in (0,1]$.

- Negative entropy $x\ln(x), x > 0$ is convex
- $\log(x)$ is concave.
- For $x \in \mathbb{R}^n$, 2-norm $\|x\|_2^2 = x^T x$ is both convex and strictly convex.

Definition 10.3.2 (proper and closed convex functions). *[5, p. 8] Let $\mathcal{X} \subseteq \mathbb{R}^n$($\mathcal{X}$ is not necessarily a convex set) and $f : \mathcal{X} \to \bar{\mathbb{R}}$ with*

$$\bar{\mathbb{R}} \triangleq \mathbb{R} \cup \{-\infty, \infty\}.$$

We have the following definitions.

- *The **effective domain** of f is*

$$dom(f) \triangleq \{x \in \mathcal{X} : f(x) < \infty\}.$$

- *The function f is said to be **proper** if*

$$f(x) > -\infty, \forall x \in \mathbb{R}^n, \text{ and } f(x) < \infty \; for \text{ some } x \in \mathbb{R}^n.$$

- *The function f is said to closed if its epigraph is closed.*

Remark 10.3.1. A affine function with nonzero slope is not a proper function.

Theorem 10.3.1 (Convexity and continuity). *(convexity implies continuity) If f is convex, then it is continuous.*

Proof. (informal) consider the epigraph a discontinuous function, then we can find out near the discontinuous point, the convexity condition will be violated. □

Theorem 10.3.2 (convexity is equivalent to convexity along all lines). *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the function $g(\alpha) = f(x + \alpha p), \alpha \in \mathbb{R}$ is convex for all $x, y \in \mathbb{R}$.*

Proof. We can easily show that

$$g(\lambda\alpha) + g((1-\lambda)\beta) \geq g(\lambda\alpha + (1-\lambda)\beta).$$

□

10.3.2 Connection to convex set

Definition 10.3.3 (epigraph). *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$, the* ***epigraph*** *[Figure 10.3.2] of f is defined as the set of points in $\mathbb{R}^n \times \mathbb{R}$:*

$$\{(x,y)|x \in \mathbb{R}^n, y \in \mathbb{R}, y \geq f(x)\}.$$

Lemma 10.3.1 (convex function and epigraph definition). *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if its epigraph is convex set.*

Proof. (forward) If we take $(x_1, t_1), (x_2, t_2) \in epi(f)$, then

$$\theta t_1 + (1-\theta)t_2 \geq \theta_1 f(x_1) + (1-\theta)f(x_2) \geq f(\theta x_1 + (1-\theta)x_2) \implies \theta(x_1, t_1) + (1-\theta)(x_2, t_2) \in epi(f).$$

(backward) Because $(x_1, f(x_1)), (x_2, f(x_2)) \in epi(f)$, by convexity of epigraph $(\theta(x_1, f(x_1)) + (1-\theta)(x_2, f(x_2)) \in epi(f)$), we have

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta_1 f(x_1) + (1-\theta)f(x_2)$$

□

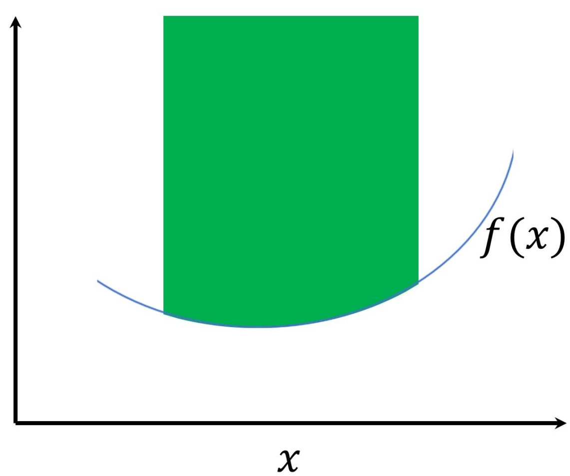


Figure 10.3.2: The epigraph (green area) of a convex function.

10.3.3 Strongly convex functions

Definition 10.3.4 (strongly convex function). *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a convex set and $f : \mathcal{X} \to \bar{\mathbb{R}}$ be a proper convex function. f is **strongly convex** if there exists $\alpha > 0$ such that*

$$f(x) - \alpha\|x\|^2$$

is a convex function.

Remark 10.3.2 (interpretation). A strongly convex function is not as flat as a regular convex function. The shape of a strongly convex function is more bowl-like.

Lemma 10.3.2 (strongly convex is strictly convex). *If f is strongly convex, then f is strictly convex and also convex.*

Proof. Based on the definition of strong convexity, we have

$$f(\lambda x + (1-\lambda)y) - \alpha\|\lambda x + (1-\lambda)y\|^2 \leq \lambda f(x) + (1-\lambda)f(y) - \lambda\alpha\|x\|^2 - (1-\lambda)\alpha\|y\|^2$$

Rearrange and we have

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - \lambda\alpha\|x\|^2 - (1-\lambda)\alpha\|y\|^2 + \alpha\|\lambda x + (1-\lambda)y\|^2.$$

Since $\|x\|^2$ is strictly convex, we have

$$\lambda\alpha\|x\|^2 + (1-\lambda)\alpha\|y\|^2 - \alpha\|\lambda x + (1-\lambda)y\|^2 > 0, \forall x, y, x \neq y, \forall \lambda \in (0,1).$$

Therefore,

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

□

10.3.4 Operations preserve convexity

Lemma 10.3.3. *[5, p. 12] The following operations preserve convexity:*

- *(addition) If f_1 and f_2 are convex functions, then $f_1 + f_2$ is convex.*
- *(maximization) If $f_1, f_2, ..., f_k$ are convex functions, then $\max(f_1, f_2, ..., f_k)$ is convex function.*
- *(composition) If $g : \mathbb{R} \to \mathbb{R}$ is non-decreasing and convex, $h : \mathbb{R}^n \to \mathbb{R}$ is convex, then $g \circ h$ is convex.*

10.3.5 Convexity and derivatives

Theorem 10.3.3 (first derivative and linear under-estimator). *Let $\mathcal{D}$ be an open set and convex set in $\mathbb{R}^n$, and let $f : \mathcal{D} \to \mathbb{R}$ be differentiable on $\mathcal{D}$. Then,*

- *f is convex on $\mathcal{D}$ if and only if*

$$y - x \geq \nabla f(y - x), \forall x, y \in \mathcal{D}$$

- *f is strictly convex on $\mathcal{D}$ if and only if above inequality is strict whenever $x \neq z$*

Proof. (1, forward)Since f is convex, we have

$$f(\lambda y + (1-\lambda)x) \leq \lambda f(y) + (1-\lambda) f(x), \forall \lambda \in [0,1], x, y \in \mathrm{dom}(f),$$

By rearranging, we have

$$\begin{aligned} f(x + \lambda(y-x)) &\leq f(x) + \lambda(f(y) - f(x)) \\ \Rightarrow f(y) - f(x) &\geq \frac{f(x + \lambda(y-x)) - f(x)}{\lambda}, \forall \lambda \in (0,1] \end{aligned}$$

As $\lambda \to 0$, we use the definition of gradient and get

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

(1, backward) Let $z = \lambda x + (1-\lambda)y$. We have

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

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

If we multiply the first by λ and the second by $(1-\lambda)$ and add together, we have

$$\begin{aligned} \lambda f(x) + (1-\lambda) f(y) &\geq f(z) + \nabla f^T(z)(\lambda x + (1-\lambda)y - z) \\ &= f(z) = f(\lambda x + (1-\lambda)y) \end{aligned}$$

(2) Similar to (1). □

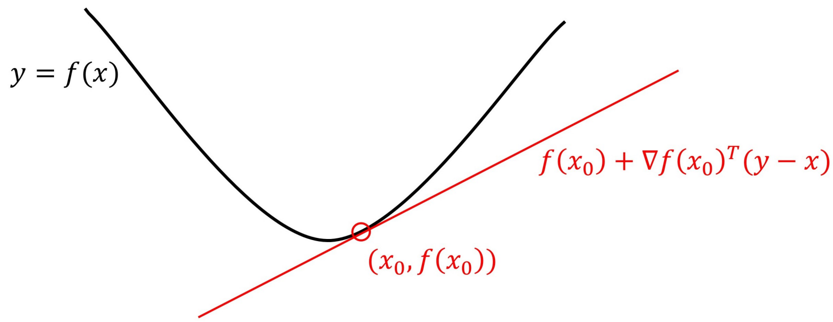


Figure 10.3.3: Illustration of linear underestimator.

Theorem 10.3.4 (convexity and second derivative). *Let $\mathcal{D}$ be an open set and convex set in $\mathbb{R}^n$, and let $f : \mathcal{D} \to \mathbb{R}$ be twice differentiable on $\mathcal{D}$. Then, f is convex on $\mathcal{D}$ if and only if*

$$\nabla^2 f \geq 0, \forall x \in \mathcal{D}.$$

Proof. (forward) We first prove the case that dimensionality is 1. Let $y > x$, then

$$f(y)\ geq f(x) + f'(x)(y-x), f(x) \geq f(y) + f'(y)(x-y).$$

We then have

$$f'(x)(y-x) \leq f(y) - f(x) \leq f'(y)(y-x).$$

Diving both sides by $(y-x)^2$ gives

$$\frac{f'(y) - f'(x)}{y - x} \geq 0, \forall x, y, x \neq y.$$

As we let $y \to x$, we get $f''(x) \geq 0$. Now we consider general dimensionality. From the zero-order condition, convexity is equivalent to convexity along all lines in the domain [Theorem 10.3.2]. Let $v \in \mathcal{D}$, then $v^T \nabla(x) v \geq 0$ indicate $\nabla^2 f \geq 0$.

(backward) Suppose $f''(x) \geq 0$. By mean value theorem, we have

$$f(y) = f(x) + f'(x)(y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x),$$

for some $z \in [x, y]$. Then we have

$$f(y) \geq f(x) + f'(x)(y-x).$$

□

Corollary 10.3.4.1 (strong convexity and second derivative). *If f is a strongly convex function, then there exists a $\sigma > 0$ such that*

$$\nabla^2 f \geq \sigma I.$$

In other words, the smallest eigenvalue of the Hessian of f is uniformly lower bounded by σ everywhere.

Lemma 10.3.4 (relation to convex function). *Suppose a differentiable function $f : X \to \mathbb{R}$ is strongly convex such that there exists $\sigma > 0$ and $f(x) - \sigma\|x\|^2$ is convex.*

Then

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\sigma}{2}\|x-y\|^2, \forall x, y \in X$$

Proof. Because $f(x) - \sigma\|x\|^2$ is convex, we have

$$f(y) - \sigma\|y\|^2 \geq f(x) - \sigma\|x\|^2 + (\nabla f(x) - 2\sigma x)^T(y-x),$$

Rearrange and we have

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\sigma}{2}\|x-y\|^2.$$

□

Note 10.3.2 (convex and strongly convex). If $f \in C^2$ and the domain is $\mathbb{R}$, then

- f is convex if and only if $f''(x) \geq 0, \forall x$.
- f is strongly convex if and only if $f''(x) \geq m > 0, \forall x$.

Note that for strongly convex function, $f''(x)$ is uniformly bounded away from 0. closely.

Example 10.3.2.

- $f(x) = x^4$ has $f''(x) = 12x^2 \geq 0$, so f is a convex function and strictly convex function. It is not strongly convex because $\{f''(x_n)\} \to 0$ for the sequence $\{1/n\}$.

10.3.6 Subgradient

Definition 10.3.5 (subgradient, subdifferential). *Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a proper convex function and $x \in dom(f)$. Any $g \in \mathbb{R}^n$ satisfying*

$$f(y) \geq f(x) + g^T(y-x), \forall y \in \mathbb{R}^n$$

*is called a **subgradient** of f at x.*

*The set of all subgradients of f at x is called **subdifferential** of f at x, denoted as $\partial f(x)$.*

Remark 10.3.3.

- If $g \in \partial f(x)$, then the epigraph of f lies above the linear underestimator
$$l(y) \triangleq f(x) + g^T(y-x).$$
- By convention, if $x \notin dom(f)$, then $\partial f(x) = \emptyset$.

Example 10.3.3. Consider $f(z) = |z|$. For $x \neq 0$, $\partial f(x) = \{1\}$; For $x = 0$, $\partial f(0) = [-1, 1]$.

Lemma 10.3.5 (basic properties of subgradient).

- *$g \in \mathbb{R}^n$ is a subgradient of $f : \mathbb{R}^n \to \mathbb{R}$ at $x_0 \in \mathbb{R}^n$ if and only if a hyperplane with normal vector $(g, -1) \in \mathbb{R}^{n+1}$ supports $epi(f)$ at $(x_0, f(x_0))$.*
- *If f is convex and differentiable, then $\nabla f(x)$ is a subgradient of f at x.*

Proof. (1) The hyperplane supporting at $(x_0, f(x_0))$ is given by the set $\{(x,y) | (g,-1)(x,y)^T = (g,-1)(x_0, f(x_0))\}$. (forward) Let (x,y) be a point in the $epi(f) = \{(x,y) | x \in \mathbb{R}^n, y \in \mathbb{R}, y \geq f(x)\}$, then we have $(g,-1)(x,y)^T \leq (g,-1)(x_0, f(x_0))$; this, the every point in $epi(f)$ is belonging to the half-space of $\{(x,y) | (g,-1)(x,y)^T \leq (g,-1)(x_0, f(x_0))\}$. Therefore, we have showed that g is subgradient by definition. since $(x, f(x))$ is in the epigraph. And g is a subgradient by definition. (backward) If g is subgradient, then

$$f(x) \geq f(x_0) + g^T(x - x_0)$$

and

$$y \geq f(x) \geq f(x_0) + g^T(x - x_0)$$

where (x,y) are in epigraph. Then, $epi(f)$ is supported by the hyperplane. (2) From Theorem 10.3.3. □

10.4 Duality theory

Definition 10.4.1 (constrained convex optimization problem). *Let $f : \mathcal{X} \to \mathbb{R}$ and $c_i : \mathcal{X} \to \mathbb{R}, i = 1, ..., m$ be convex functions, and let $\mathcal{X} \subseteq \mathbb{R}^n$ be a nonempty convex set. We consider the following convex optimization problem.*

$$\min_{x\in\mathbb{R}^n} f(x), \text{ subject to } x \in \mathcal{X} = \{x : c(x) \leq 0\}$$

where

$$c(x) = \begin{bmatrix} c_1(x) \\ c_2(x) \\ \cdots \\ c_m(x) \end{bmatrix}.$$

Definition 10.4.2 (primal function and dual function). *For convex optimization problem [Definition 10.4.1], define the Lagrangian $\mathcal{L} : \mathcal{X} \times \mathbb{R}^m \to \mathbb{R}$ as*

$$\mathcal{L}(x, y) = f(x) + c(x)^T y = f(x) + \sum_{i=1}^{m} c_i(x) y_i$$

*where $y \in \mathbb{R}^m, y \geq 0$ is called dual variables and $x \in \mathcal{X}$ is called primal variables. The **primal** function $p(x)$ is defined as*

$$p(x) = \begin{cases} \sup_{y\in\mathcal{Y}} \mathcal{L}(x, y), \text{ if } x \in \mathcal{X} \\ \infty, \text{ if } x \notin \mathcal{X} \end{cases},$$

*and the **dual function** is defined as*

$$d(y) = \begin{cases} \inf_{x\in\mathcal{C}} \mathcal{L}(x, y), \text{ if } y \in \mathcal{Y} \\ -\infty, \text{ if } y \notin \mathcal{Y} \end{cases},$$

where $\mathcal{Y} = \{y : y_i \geq 0, \forall i = 1, 2, ..., m\}$.

Definition 10.4.3 (primal problem and dual problem). *The **primal problem** is defined as*

$$\min_{x\in\mathbb{R}^n} p(x) = \min_{x\in\mathcal{X}}\{\sup_{y\in\mathcal{Y}} \mathcal{L}(x,y)\}.$$

*The **dual problem** is defined as*

$$\max_{y\in\mathbb{R}^m} d(y) = \max_{y\in\mathcal{Y}}\{\inf_{x\in\mathcal{X}} \mathcal{L}(x,y)\}.$$

Lemma 10.4.1 (primal function "is" original function). *The primal function is equivalent to*

$$p(x) = \begin{cases} f(x), \textit{if } x\in\mathcal{X}, c(x)\leq 0 \\ \infty, \textit{otherwise} \end{cases}$$

Proof. When $c(x) \leq 0$, y will take 0 to maximize $\mathcal{L}(x,y)$ in y; that is $p(x) = \sup_{y\in\mathcal{Y}} \mathcal{L}(x,y) = \mathcal{L}(x,0) = f(x)$. When there exist one $c_i(x) > 0$, $p(x) = \sup_{y\in\mathcal{Y}} = \infty$. □

Remark 10.4.1 (dual function is explicit only for simple cases)**.** Note that dual function has explicit form only for simple optimization problems, such as linear optimization.

Lemma 10.4.2 (The dual function is a concave function). *The dual function defined as*

$$d(y) = \begin{cases} \inf_{x\in\mathcal{C}} \mathcal{L}(x,y), \textit{ if } y\in\mathcal{Y} \\ -\infty, \textit{ if } y\notin\mathcal{Y} \end{cases},$$

where $\mathcal{Y} = \{y : y_i \geq 0, \forall i = 1,2,...,m\}$, is a concave function.

Proof.

$$\begin{aligned} d(\lambda y_1 + (1-\lambda)y_2) &= \inf_x \mathcal{L}(x, \lambda y + (1-\lambda)y) \\ &= \inf_x \mathcal{L}(x,\lambda y) + \mathcal{L}(x,(1-\lambda)y) \\ &\geq \inf_x \mathcal{L}(x,\lambda y) + \inf_x \mathcal{L}(x,(1-\lambda)y) \\ &\geq d(\lambda y_1) + d((1-\lambda)y_2)c. \end{aligned}$$

□

Example 10.4.1 (linear programming). [6, lec 2] Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$ and consider the linear program

$$\min_{x\in\mathbb{R}^n} f(x) = g^T x, \text{ subject to } Ax - b \leq 0$$

and assume that $Ax - b \leq 0$ is feasible. The Lagrangian and dual functions are given by

$$\mathcal{L}(x,y) = g^T x + (Ax - b)^T y$$

and

$$d(y) = \begin{cases} \inf_{x\in\mathbb{R}^n} g^T x + (Ax-b)^T y = \inf_{x\in\mathbb{R}^n} x^T(g + A^T y) - b^T y, if\ y \geq 0 \\ -\infty, otherwise \end{cases}$$

Note that for linear optimization $\inf_{x\in\mathbb{R}^n} x^T(g + A^T y) - b^T y$, if $g + A^T y \neq 0$, then $\inf_{x\in\mathbb{R}^n} x^T(g + A^T y) - b^T y = -\infty$. Therefore, we can further simplify the dual function to

$$d(y) = \begin{cases} -b^T y, if\ y \geq 0, g + A^T y = 0 \\ -\infty, otherwise \end{cases}$$

Maximize the dual function can be written as

$$\max_{y\in\mathbb{R}^m} -b^T y, subject\ to\ g + A^T y = 0, y \geq 0.$$

Theorem 10.4.1 (weak duality I). *For any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, we have*

$$d(y) \leq p(x).$$

Moreover, if x is a primal feasible point, then

$$d(y) \leq f(x).$$

Let $f_{opt} = \inf_{x\in\mathcal{X}} f(x)$, subject to $x \in \mathcal{X}, c(x) \leq 0$. If it is feasible, then

$$\sup_{d\in\mathbb{R}^m} d(y) \leq \inf_{x\in\mathcal{X}} f(x) < \infty.$$

Proof. (1)

$$d(y) = \inf_x L(x,y) \leq L(x,y) \leq \sup_y L(x,y) = p(x)$$

(2) When x is primal feasible, then $p(x) = f(x)$. □

Definition 10.4.4 (strong duality). *If the equality $d(y^*) = p(x^*)$ holds, we say the* ***strong duality*** *holds. However, strong duality does not always hold. For certain type of constraints, only some simple conditions need to be satisfied to ensure strong duality.*

Consider convex optimization of the form minimize $\quad f_0(x)$ subject to $g_i(x) \leq 0, \quad i = 1, \ldots, m \; Ax = b$ where $f, g_1, ...g_m$ are convex functions.

Then if there exists an $x \in relintD$ such that non-affine inequalities are strictly satisfied, i.e., $g_i(x) < 0, i = 1, ..., m$, $Ax = b$, which is the so-called **Slater's condition**, then strong duality holds.

10.5 Convex optimization and optimality conditions

10.5.1 Local optimality vs. global optimality

Lemma 10.5.1 (local minimum implies global minimum). *If X is a convex subset of $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is convex over X, then a local minimum of f over X is also a global minimum.*

Proof. let x^* be a local minimum, suppose there is a point $x' \neq x^*$ being a global minimum. Then

$$f(x^* + a(x' - x^*)) \leq af(x^*) + (1-a)f(x') < af(x^*) + (1-a)f(x^*) < f(x^*)$$

for $a \in [0,1]$ which contradicts that x^* is local minimum. □

Note: The domain of convex function has to be convex, which is the definition of convex functions on a general set. Otherwise, for $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$, the element $(\lambda x + (1-\lambda)y)$

Lemma 10.5.2 (uniqueness of global minimum under strict convexity). *[4, p. 17] If X is a convex subset of $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is **strictly convex** over X, then f over X has at most one global minimum. In other words, **the set of minimizers is either empty or a singleton.***

Proof. Suppose there are two global minimum at x_1, x_2, then the middle $(x_1 + x_2)/2$ will have a smaller value based on the definition of **strict convexity**. □

Remark 10.5.1 (global minimizer might not exist). Consider $f(x) = e^{-x}$. It is strictly convex since $f''(x) = e^{-x}$. The global minimizer is $-\infty \notin \mathbb{R}$. However, for strongly convex function, the global minimizer will exist on closed convex set.

Lemma 10.5.3 (existence and uniqueness of global minimizer under strongly convexity). *[4, p. 17] If X is a **convex and closed** subset of $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is **strongly convex** over X, then f over X has one and only one global minimum.*

Proof. Note that from the properties of strongly convex function [subsection 10.3.3], it can be showed that it is a coercive function. A coercive function on a closed set has a global minimizer [Theorem 3.1.7]. Moreover, strongly convex function is also strictly convex function, therefore, the minimizer is unique. □

10.5.2 Unconstrained optimization optimality conditions

Theorem 10.5.1 (necessary and sufficient optimality condition, differentiable function). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex **differentiable** function and consider the problem*

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

Then x^* is a global minimizer if and only if

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

Proof. Let $g \triangleq \partial f(x^*)$, then from the property of first derivative and convexity [Theorem 10.3.3]

$$f(y) \geq f(x^*) + g^T(y - x^*), \forall y \in \mathcal{X}.$$

Take $g = 0$, then

$$f(y) \geq f(x^*), \forall y \in \mathcal{X}.$$

□

Theorem 10.5.2 (necessary and sufficient optimality condition via subdifferential, nonsmooth function). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and consider the problem*

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

Then x^* is a global minimizer if and only if

$$0 \in \partial f(x^*).$$

Proof. Let $g \in \partial f(x^*)$, then based on the definition of subgradient we have

$$f(y) \geq f(x^*) + g^T(y - x^*), \forall y \in \mathcal{X}.$$

Take $g = 0$, then

$$f(y) \geq f(x^*), \forall y \in \mathcal{X}.$$

□

10.5.3 Constrained optimization optimality conditions

Definition 10.5.1 (general constrained convex optimization). *The convex constrained optimization problem is given as*

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

for some nonempty set $\mathcal{X} \subseteq \mathbb{R}^n$ *and function* $f : \mathbb{R}^n \to (-\infty, \infty]$

Theorem 10.5.3 (necessary and sufficient condition). *[4, p. 17]Let* X *be a convex set and let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a convex function over* X*.*

1. *If* f *is continuously differentiable, then*

$$\nabla f(x^*)^T (x - x^*) \geq 0, \forall x \in X$$

is a necessary and sufficient condition for x^* *to be a global minimum of* f *over* X*.*

2. *If* X *is open and* f *is continuously differentiable, then*

$$\nabla f(x^*) = 0, \forall x \in X$$

is a necessary and sufficient condition for x^* *to be a global minimum of* f *over* X

Proof. (1) (a)From $f(x) - f(x^*) \geq f(x^*)(x - x^*)$ and $f(x^*)(x - x^*) \geq 0$, we have

$$f(x) \geq f(x^*), \forall x \in X$$

; (b) If $f(x^*)(x - x^*) < 0$, then

$$(f(x + a(x^* - x)) - f(x)) = \nabla a f(x^*)^T (x - x^*) + o(a(x - x^*)) < 0$$

, as $a > 0, a \to$, therefore, we can decrease the function, which is a contraction. (2)(a) If $\nabla f(x^*) = 0$, from (1) we can prove the forward direction (b) suppose $\nabla f(x^*) > 0$, then we can decrease it by move in $-\nabla f(x^*)$ in sufficiently small steps. □

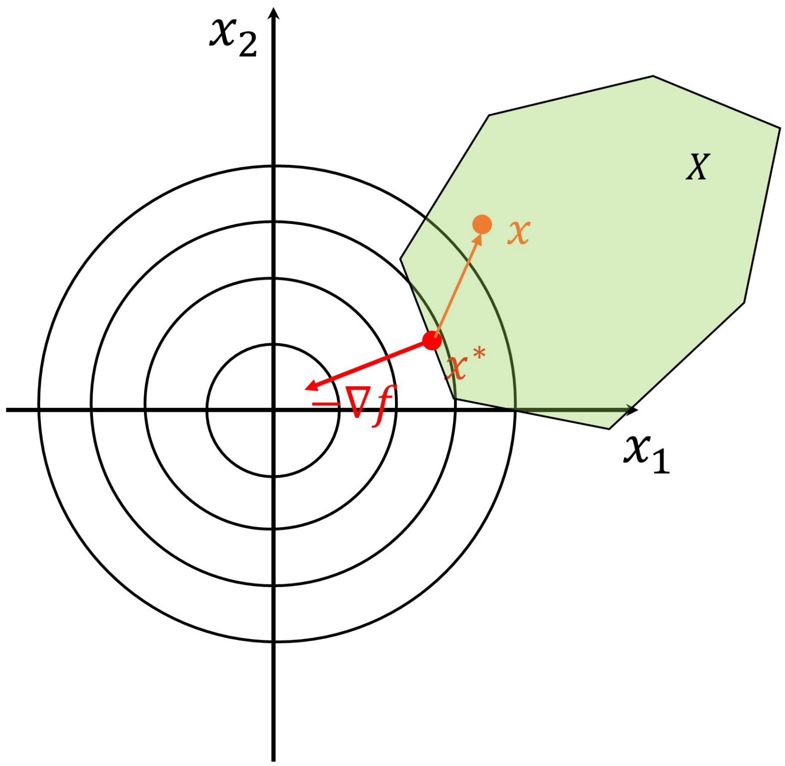


Figure 10.5.1: Demonstration of optimality condition $\nabla f(x^*)^T(x - x^*) \geq 0, \forall x \in X$ when x^* lies on the boundary of X.

Remark 10.5.2.

- If X is not a open set, then we cannot guarantee there is a neighborhood around x^*, and therefore we have to use the first item, because $\nabla f(x^*)$ might not exist if x^* is at the boundary of X.
- No matter X is open or not, as long as X is convex, we can always use the first item.
- For unconstrained optimization on $\mathbb{R}^m$, we usually use the the second item because $\mathbb{R}^m$ is considered open.

Remark 10.5.3 (Why we do not need second order condition)**.**

- We do not need second order necessary condition because if f is twice differentiable, then $\nabla f \geq 0$ as implies by convexity of f.
- We do not need second order sufficient condition like because $\nabla f > 0$ if f is twice differentiable because we rely on the condition $f(y) \geq f(x) + \nabla f(x)^T(y - x)$ in the proof, which is a much stronger condition.

- The second order sufficient condition $\nabla f > 0$ in general nonlinear optimization is a 'strong' condition that directly guarantees the existence of 'strict' local minimum, which does not direct counterpart in convex optimization. In convex optimization, if we want 'strict' local minimum, we certainly need 'extra information/conditions'.

Theorem 10.5.4 (KKT condition, inequality constraint). *Consider an convex optimization problem given by*

$$\begin{aligned} &\min \quad f(x) \\ &\textit{subject to } h_i(x) \leq 0, i = 1, \dots m \\ &\ell_j(x) = 0, j = 1, \dots r \end{aligned}$$

*where $f(x), h_1, ..., h_m$ are convex function, $l_1, ..., l_r$ are linear equality constraints. The **Karush-Kuhn-Tucker conditions (KKT conditions)** are:*

- *stationarity:* $0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial \ell_j(x)$
- *complementary:* $u_i h_i = 0, \forall i$
- *primal feasibility:* $h_i(x) \leq 0, l_j(x) = 0, \forall i, j$
- *dual feasibility:* $u_i \geq 0, \forall i$

Assume strong duality holds, then there exists x^*, u^*, v^* satisfying the KKT condition if and only if x^*, u^*, v^* are primal and dual solutions.

Proof. (forward)

Note that

$$\begin{aligned} f\left(x^*\right) &= g\left(u^*, v^*\right) \\ &= \min_x f(x) + \sum_{i=1}^{m} u_i^* h_i(x) + \sum_{j=1}^{r} v_j^* \ell_j(x) \\ &\leq f\left(x^*\right) + \sum_{i=1}^{m} u_i^* h_i\left(x^*\right) + \sum_{j=1}^{r} v_j^* \ell_j\left(x^*\right) \\ &\leq f\left(x^*\right) \end{aligned}$$

Therefore above equality holds. We must have

- x^* minimizes $L\left(x, u^*, v^*\right)$ over x, i.e.,

$$\begin{aligned} &0 \in \partial_x L\left(x^*, u^*, v^*\right) \\ &0 \in \partial f\left(x^*\right) + \sum u_i^* \partial h_i\left(x^*\right) + \sum v_j^* \partial \ell_j\left(x^*\right) \end{aligned}$$

which is the stationary condition.

- $\sum u_i^* h_i(x^*) = 0$, i.e.
$$u_i^* h_i(x) = 0 \text{ for all } i$$
which is the complementary slackness condition.
- The primal and dual feasibility of (x^*, u^*, v^*) hold.

(backward) If there exists x^*, u^*, v^* that satisfy the KKT conditions, then

$$\begin{aligned} g(u^*, v^*) &= f(x^*) + \sum_{i=1}^{m} u_i^* h_i(x^*) + \sum_{j=1}^{r} v_j^* \ell_j(x^*) \\ &= f(x^*) \end{aligned}$$

where the first equality holds from stationarity, and the second holds from complementary slackness. Therefore, the duality gap is zero, so x^* and u^*, v^* are primal and dual optimal respectively. □

Remark 10.5.4 (compare with KKT conditions in nonlinear constrained optimization).

- For convex optimization with inequality constraints, we only need first order condition, whereas general nonlinear optimization will require second order condition.

Example 10.5.1. Consider the support vector machine optimization problem given by

$$\begin{aligned} \min_{\beta\in\mathbb{R}^p, \beta_0\in\mathbb{R}, \xi\in\mathbb{R}} \quad & \tfrac{1}{2}\|\beta\|_2^2 + C\textstyle\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & \xi_i \geq 0, i = 1, \cdots, n \\ & y_i\left(x_i^T\beta + \beta_0\right) \geq 1 - \xi_i, i = 1, \cdots, n \end{aligned}$$

where $x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}$. Denote the dual variables by $v, w \geq 0$. The KKT condition gives

$$0 = \sum_{i=1}^{n} w_i y_i, \quad \beta = \sum_{i=1}^{n} w_i y_i x_i, \quad w = C\mathbf{1} - v$$

and complementary slackness condition:

$$v_i \xi_i = 0, w_i\left(1 - \xi_i - y_i\left(x_i^T\beta + \beta_0\right)\right) = 0, \quad i = 1, \cdots, n.$$

10.6 Subgradient methods

10.6.1 A generic algorithm for unconstrained problem

We first consider minimizing an unconstrained function $f(x)$, which is convex and defined on $\mathbb{R}^n$. Subgradient methods employ a similar idea in gradient descent, with gradients replaced by subgradient. Starting from an iterate $x^{(0)}$, we update the iterate via

$$x_k = x_{k-1} - \alpha_k \cdot g_{k-1}, k = 1, 2, 3, \cdots$$

Compared to general gradient methods, the step size α_k can be chosen in a much simpler way and still have good convergence property. Most common two ways are:

- Fixed step size: $\alpha_k = s$ for all $k = 1, 2, 3 \cdots$
- Diminishing step size: choose α_k to satisfy

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty \, , \; \sum_{k=1}^{\infty} \alpha_k = \infty$$

A generic subgradient algorithm is in the following algorithm 16.

Algorithm 16: A generic subgradient algorithm

Input: Initial guess $x_0 \in \mathbb{R}^n$
Set $k = 0$.
repeat
  Obtain a subgradient $g_k \in \partial f(x_k)$
  if $g_k = 0$ **then**
    return a global minimizer x_k.
  end
  Choose $\alpha_k > 0$
  Set $x_{k+1} = x_k - \alpha_k g_k$.
  Record the best iterate so far x_k^{best} that has the minimum $f(x)$.
  set $k = k + 1$.
until *termination condition*;

Theorem 10.6.1. *Consider the subgradient algorithm [algorithm 16]. Given a convex function $f : \mathbb{R}^n \to \mathbb{R}$ that satisfies:*

- *f has its subgradient bounded by $\|g\|_2 < G$.*
- *$||x_0 - x^*|| \leq R$ which means it is bounded.*

Then we have

- *For a fixed step size $\alpha_k = s$, we have*

$$\lim_{k\to\infty} f(x_k^{best}) \le f(x^*) + \frac{G^2 s}{2}$$

- *For a step size α_k satisfying $\sum_{k=1}^{\infty}\alpha_k^2 < \infty, \sum_{k=1}^{\infty}\alpha_k < \infty$, we have*

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

Proof.

$$\begin{aligned}||x_{k+1} - x^*||^2 =& ||x_k - t_k g_k - x^*||^2 \\ =& ||x_k - x^*||^2 - 2t_k(g_k)^T(x_k - x^*) + \alpha_k^2||g_k||^2\end{aligned}$$

By definition of the subgradient method, we have

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

Using this inequality, we have

$$||x_{k+1} - x^*||^2 \le ||x_k - x^*||^2 - 2t_k(f(x_k) - f(x^*)) + \alpha_k||g_k||^2.$$

Sum up k terms, we can arrive at

$$||x_{k+1} - x^*||^2 \le ||x_1 - x^*||^2 - 2\sum_{i=1}^{k} t_i(f(x_i) - f(x^*)) + \sum_{i=1}^{k}\alpha_i^2||g_i||^2.$$

Then we have

$$0 \le ||x_{k+1} - x^*||^2 \le R^2 - 2\sum_{i=1}^{k}\alpha_i(f(x_i) - f(x^*)) + \sum_{i=1}^{k}\alpha_i^2 G^2$$

$$2\sum_{i=1}^{k}\alpha_i(f(x^{(i)}) - f(x^*)) \le R^2 + \sum_{i=1}^{k}\alpha_i^2 G^2$$

$$2(\sum_{i=1}^{k}\alpha_i)(f(x_k^{best}) - f(x^*)) \le R^2 + \sum_{i=1}^{k}\alpha_i^2 G^2$$

For a constant step size $\alpha_i = s$:

$$\frac{R^2 + G^2 s^2 k}{2sk} \to \frac{G^2 s}{2}, \text{ as } k \to \infty,$$

and for diminishing step size, we have:

$$\sum_{i=0}^{k} \alpha_i^2 \leq 0, \sum_{i=0}^{k} \alpha_i = \infty$$

therefore,

$$\frac{R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2 \sum_{i=0}^{k} \alpha_i} \to 0, \text{ as } k \to \infty,$$

□

Remark 10.6.1 (estimation of convergence speed). If we take $\alpha_i = R/(G\sqrt{k})$, for all $i = 1, ..., k$. Then we can obtain the following bound:

$$\frac{R^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2 \sum_{i=0}^{k} \alpha_i} = \frac{R^2 + G^2}{\sqrt{k}}.$$

That is, subgradient method has convergence rate of $O(1/\sqrt{k})$, and to get $f(x_{best}^{(k)}) - f(x^*) \leq \epsilon$, needs $O(1/\epsilon^2)$ iterations.

10.6.2 Convergence under Lipschitz smoothness

Definition 10.6.1 (Lipschitz continuous function). *A function f is $L-$smoothness if there exists a constant $L \geq 0$ such that*

$$\left\|\nabla f(x) - \nabla f(y)\right\|_2 \leq L\left\|x - y\right\|_2, \forall x, y \in \mathcal{X}.$$

Example 10.6.1. Consider a quadratic function

$$f(x) = \frac{1}{2}x^T Bx + c^T x, x \in \mathbb{R}^n.$$

We have [Theorem 5.13.1]

$$\nabla f(x) = Bx + c, \left\|\nabla f(x) - \nabla f(y)\right\|_2 = \left\|B(x - y)\right\| \leq \|B\|_2 \left\|x - y\right\|.$$

Therefore, if the max magnitude of eigenvalues, i.e., $\max_{1 \leq i \leq n} |\lambda_i|$ equals L, then f has a Lipschitz continuous gradient constant L.

Lemma 10.6.1 (properties of Lipschitz continuous function). *Suppose f is convex and L -smooth. We have*

- $$\frac{1}{2L}\|\nabla f(x)-\nabla f(y)\|_2^2 \leq f(y)-f(x)-\nabla f(x)^T(y-x) \leq \frac{L}{2}\|x-y\|_2^2, \forall x,y \in \mathcal{X}$$
- $$f(y) \geq f(x)+\nabla f(x)^T(y-x)+\frac{1}{2L}\|\nabla f(x)-\nabla f(y)\|_2^2, \forall x,y \in \mathcal{X}$$
- $$[\nabla f(x)-\nabla f(y)]^{\dot{T}}(x-y) \geq \frac{1}{L}\|\nabla f(x)-\nabla f(y)\|_2^2 \forall x,y \in \mathcal{X}$$

Proof. (1)

$$\begin{aligned}
& f(y)-f(x)-\nabla f(x)^T(y-x) \\
& =\int_0^1 \nabla f(x+t(y-x))^T(y-x)dt-\nabla f(x)^T(y-x) \\
& =\int_0^1 [\nabla f(x+t(y-x))-\nabla f(x)]^T(y-x)dt \\
& \leq \int_0^1 \|\nabla f(x+t(y-x))-\nabla f(x)\|_2\|y-x\|_2 dt \text{ (by Cauchy-Schwarz inequality)} \\
& \leq \int_0^1 L\left[t(y-x)\right\|_2\| y-x\|_2 dt \text{ (by L-smoothness of } f) \\
& =L\|(y-x)\|_2^2\int_0^1 tdt \\
& =\frac{L}{2}\|(y-x)\|_2^2
\end{aligned}$$

(2) Let $z=y+\frac{1}{L}(\nabla f(x)-\nabla f(y))$ We have

$$\begin{aligned}
f(y)-f(x) &= f(y)-f(z)+f(z)-f(x) \\
&\geq -\nabla f(y)^T(z-y)-\frac{L}{2}\|y-z\|_2^2+\nabla f(x)^T(z-x) \\
&= \nabla f(x)^T(y-x)-\{\nabla f(x)-\nabla f(y)\}^T(y-z)-\frac{L}{2}\|y-z\|_2^2 \\
&= \nabla f(x)^T(y-x)+\frac{1}{L}\|\nabla f(x)-\nabla f(y)\|_2^2-\frac{1}{2L}\|\nabla f(y)-\nabla f(x)\|_2^2 \\
&= \nabla f(x)^T(y-x)+\frac{1}{2L}\|\nabla f(x)-\nabla f(y)\|_2^2
\end{aligned}$$

(3) Add the following two together.

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2L}\|\nabla f(x) - \nabla f(y)\|_2^2$$
$$f(x) \geq f(y) + \nabla f(y)^T(x-y) + \tfrac{1}{2L}\|\nabla f(y) - \nabla f(x)\|_2^2$$

□

Lemma 10.6.2 (reduction per subgradient step). *Let f be convex and L -smooth. The subgradient descent step strictly reduces the objective function value at each iteration,*

$$f\left(x_{k+1}\right) - f\left(x_k\right) \leq -\alpha\left(1 - \frac{L}{2}\alpha\right)\left\|\nabla f\left(x_k\right)\right\|_2^2$$

If we take $\alpha = \frac{1}{L}$, we get the maximum reduction, which is given by

$$f\left(x_{k+1}\right) - f\left(x_k\right) \leq -\frac{1}{2L}\left\|\nabla f\left(x_k\right)\right\|_2^2.$$

Proof.

$$\begin{aligned} f\left(x_{t+1}\right) - f\left(x_t\right) &\leq \nabla f\left(x_t\right)^T\left(-\alpha \nabla f\left(x_t\right)\right) + \frac{L}{2}\left\|-\alpha \nabla f\left(x_t\right)\right\|_2^2 \\ &= -\alpha\left(1 - \frac{L}{2}\alpha\right)\left\|\nabla f\left(x_t\right)\right\|_2^2 \end{aligned}$$

□

Theorem 10.6.2 (convergence of subgradient methods on Lipschitz continuous function). *[7] Let f be convex and L -smooth, x^* be an optimal solution. With $\alpha = \frac{1}{L}$ for all $\alpha > 0$, the iterates from subgradient descent satisfies*

$$f\left(x_k\right) - \min_{x \in \mathbb{R}^n} f(x) \leq \frac{2L\left\|x_0 - x^*\right\|_2^2}{k}$$

Proof. Note that we have

- $f\left(x_{k+1}\right) - f\left(x_k\right) \leq -\frac{1}{2L}\left\|\nabla f\left(x_k\right)\right\|_2^2$.

- $\|x_{k+1} - x_*\|_2 \leq \|x_k - x_*\|_2$ This is because

$$\begin{aligned}
\|x_{k+1} - x^*\|_2^2 &= \left\|x_k - \frac{1}{L}\nabla f(x_t) - x^*\right\|_2^2 \\
&= \|x_k - x^*\|_2^2 - \frac{2}{L}\nabla f(x^*)^T (x_t - x^*) + \frac{1}{L^2}\|\nabla f(x_t)\|_2^2 \\
&\leq \|x_k - x^*\|_2^2 - \frac{1}{L^2}\|\nabla f(x_k)\|_2^2 \\
&\leq \|x_k - x^*\|_2^2
\end{aligned}$$

- $\|\nabla f(x_i)\|_2 \geq \frac{f(x_i)-f(x_i)}{\|x_i-x_i\|}$
  because

$$f(x_k) - f(x^*) \leq \nabla f(x_k)^T (x_k - x_k) \leq \|\nabla f(x_k)\|_2 \|x_k - x^*\|_2$$

By combining them, we arrive at

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\left[\frac{f(x_k) - f(x_*)}{\|x_0 - x_*\|_2}\right]^2$$

and set $\epsilon_k = f(x_k) - f(x_k)$ and $\beta = \frac{1}{2L_1 x_0 - x_0 + 1/3}$

$$\begin{aligned}
[f(x_{k+1}) - f(x^*)] - [f(x_k) - f(x^*)] = \epsilon_{k+1} - \epsilon_k \leq -\tfrac{1}{2L}\tfrac{\epsilon_k^2}{\|x_0 - x_*\|_2^2} = -\beta\epsilon_k^2 \\
\frac{1}{\epsilon_k} - \frac{1}{\epsilon_{k+1}} \leq -\beta\frac{\epsilon_k}{\epsilon_{k+1}} \leq -\beta \\
\Rightarrow \frac{1}{\epsilon_k} + \beta \leq \frac{1}{\epsilon_{k+1}} \\
\Rightarrow \frac{1}{\epsilon_0} + \beta k \leq \frac{1}{\epsilon_k} \\
\Rightarrow \beta k \leq \frac{1}{\epsilon_t} = \frac{1}{f(x_k) - f(x^*)}
\end{aligned}$$

which implies $f(x_k) - f(x^*) \leq \frac{2L\|x_0 - x.\|_2^2}{k}$ □

10.6.3 Projected gradient methods

10.6.3.1 *Foundations*

Now we address convex optimization problems over convex sets. The target problem is

$$\min_{x \in \mathcal{X}} f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is **continuously differentiable** and $\mathcal{X} \subseteq \mathbb{R}^n$ is a closed convex set.

Definition 10.6.2 (gradient projection arc). *[6] For any $x_k \in \mathcal{X}$, the **gradient projection arc** is defined as the set of vectors*

$$\{x_k(\alpha) : \alpha > 0\},$$

where $x_k(\alpha) \triangleq Proj_{\mathcal{X}}(x_k - \alpha \nabla f(x_k))$ and

$$Proj_{\mathcal{X}}(y) \triangleq \arg\min_{x \in \mathcal{X}} \|x - y\|_2 .$$

Lemma 10.6.3 (descent properties of gradient projection arc). *[8, p. 304][6] If $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and $\mathcal{X} \subseteq \mathbb{R}^n$ is a closed and convex set, then the following properties hold:*

- *If $x_k(\alpha) \neq x_k$ for any $\alpha > 0$, then $d_k(\alpha) \triangleq x_k(\alpha) - x_k$ is a feasible direction satisfying*

$$\nabla f(x_k)^T (x_k(\alpha) - x_k) \leq -\frac{1}{\alpha} \|x_k(\alpha) - x_k\|_2^2 < 0, \forall \alpha > 0.$$

  *That is, $\nabla f(x_k)^T d_k(\alpha) < 0$, $d_k(\alpha)$ **is descent direction for all** $\alpha > 0$ **such that** $x_k(\alpha) \neq x_k$*
- *If $x_k(\alpha) = x_k$ for some $\alpha > 0$, then x_k is first-order minimizer for the target problem.*

Proof. (1) From the obtuse angle theorem [Theorem 10.2.2],we have

$$(x_k - \alpha \nabla f(x_k) - x_k(\alpha))^T (x - x_k(\alpha)) \leq 0, \forall x \in \mathcal{X}.$$

Choose $x = x_k$, we obtain

$$(x_k - -\alpha \nabla f(x_k) - x_k(\alpha))^T (x_k - x_k(\alpha)) \leq 0.$$

Rearrange, we have

$$\nabla f(x_k)^T (x_k(\alpha) - x_k) \leq -\frac{1}{\alpha} \|x_k(\alpha) - x_k\|_2^2 < 0, \forall \alpha > 0.$$

(2) From the obtuse angle theorem [Theorem 10.2.2],we have

$$(x_k - \alpha \nabla f(x_k) - x_k(\alpha))^T (x - x_k(\alpha)) \leq 0, \forall x \in \mathcal{X}.$$

If there exists $\alpha > 0$ such that $x_k(\alpha) = x_k$, then

$$\alpha \nabla f(x_k)^T (x - x_k(\alpha)) \geq 0, \forall \alpha > 0.$$

This is exactly the first-order optimality condition [Theorem 10.5.3]. □

Theorem 10.6.3 (convergence with constant step size). *[6] If*

- *$f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function*
- *∇f is Lipschitz continuous with constant L*
- *$\mathcal{X} \subseteq \mathbb{R}^n$ be a closed and convex set*
- *$\alpha \in (0, 2/L)$.*
- *The sequence $\{x_k\}$ is generated by*

$$x_{k+1} = Proj_{\mathcal{X}}(x_k - \alpha \nabla f(x_k)).$$

Then,

- ***monotonically decreasing along** x_k:*

$$f(x_{k+1}) \leq f(x_k) - (\frac{1}{\alpha} - \frac{L}{2})\|x_{k+1} - x_k\|_2^2.$$

- *Every limit point of $\{x_k\}$ is a **first order solution**.*

10.6.3.2 *Algorithms*

Algorithm 17: Gradient projection algorithm with constant step size

Input: Input any $x_{in} \in \mathbb{R}^n$

Set $x_0 = Proj_{\mathcal{X}}(x_{in})$ such that x_0 is feasible and set $k = 0$.
Choose $\epsilon > 0$, $\alpha_0 > 0$, and $\eta_d \in (0, 1)$.
repeat
  compute next iterate via:

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

  while $f(x_{k+1}) > l(x_{k+1}; x_k) + \frac{1}{2\alpha_k}\|x_{k+1} - x_k\|^2$ **do**
    Set $\alpha_k = \eta_d \alpha_k$.
    Compute $x_{k+1} = Proj_{\mathcal{X}}(x_k - \alpha_k \nabla f(x_k))$.
  end
  Set $\alpha_{k+1} = \alpha_k$.
  set $k = k + 1$.
until *termination condition*;

In practice, if we do not know L, then how to choose $\alpha > 0$. We have select step size according to [6]:

$$f\left(x_k(\alpha)\right) \leq \ell\left(x_k(\alpha); x_k\right) + \frac{1}{2\alpha}\left\|x_k(\alpha) - x_k\right\|_2^2$$

Consequently, if we decrease α by some factor every time that holds, eventually we will have $\alpha \in (0, 1/L]$ and it will no longer need to be changed.

Algorithm 18: Gradient projection algorithm with adaptive size

Input: Input any $x_{in} \in \mathbb{R}^n$

Set $x_0 = Proj_{\mathcal{X}}(x_{in})$ such that x_0 is feasible and set $k = 0$.
Choose $\epsilon > 0$, $\alpha_0 > 0$, and $\eta_d \in (0, 1)$.
repeat
  compute next iterate via:

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

  while $f(x_{k+1}) > l(x_{k+1}; x_k) + \frac{1}{2\alpha_k}\|x_{k+1} - x_k\|^2$ **do**
    Set $\alpha_k = \eta_d \alpha_k$.
    Compute $x_{k+1} = Proj_{\mathcal{X}}(x_k - \alpha_k \nabla f(x_k))$.
  end
  Set $\alpha_{k+1} = \alpha_k$.
  set $k = k + 1$.
until *termination condition*;

Remark 10.6.2.

- Choose $\alpha_0 \approx 1/L$ if an estimate of the Lipschitz constant L is available.
- Reasonable stopping condition

$$\|x_{k+1} - x_k\|_\infty \leq \epsilon \max(1, \|x_0\|_\infty)$$

  for some small $\epsilon \approx 10^{-6}$.
- We cannot use $\|\nabla f(x_k)\| \approx 0$ as the termination condition, because we are doing constrained optimization.

10.6.4 Proximal gradient methods

10.6.4.1 *Foundations*

We consider convex optimization whose objective function can be written in two parts, which is given by

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

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is **differentiable convex function**.

- ∇f is Lipschitz continuous with constant L.
- $r : \mathbb{R}^n \to \bar{\mathbb{R}}$ is a closed proper convex function.

Different choices of $r(x)$ can be

- $r(x) = \delta_{\mathcal{X}}(x)$ will give the constrained optimization on f, and the proximal gradient projection iteration is equivalent to project gradient iteration.
- $r(x) = \lambda \|x\|_1$ gives the sparsity problem.See subsubsection 10.6.4.3.
- $r(x) = \lambda \|x\|_2$ gives the following iteration:

$$x_{k+1} = (\frac{1}{1+\alpha_k}) z_k.$$

The method proximal gradient method will carry out iteration given by

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} f(x_k) + \nabla f(x_k)^T (x - x_k) + r(x) + \frac{1}{2\alpha_k} \|x - x_k\|_2^2.$$

The proximal gradient method iteration is equivalent to a two step iteration:

$$\begin{aligned} z_k &= x_k - \alpha \nabla f(x_k) \\ x_{k+1} &= r(x) + \frac{1}{2\alpha_k} \|x - z_k\|_2^2. \end{aligned}$$

10.6.4.2 *Algorithms*

Algorithm 19: Proximal gradient algorithm

Input: Input any $x_0 \in \mathbb{R}^n$. Objective function with Liptchize constant L

Set $k = 0$, and choose $\epsilon > 0, \alpha_0 < 1/L$, and $\eta_d \in (0, 1)$

repeat

  Set $z_k = x_k - \alpha_k \nabla f(x_k)$.

  Compute next iterate via:

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^n} r(x) + \frac{1}{2\alpha_k} \|x - x_k\|_2^2.$$

  set $k = k + 1$.

until *termination condition*;

Theorem 10.6.4 (convergence of proximal gradient method). *[6] Assume*

- *$f : \mathbb{R}^n \to \mathbb{R}$ is a convex differentiable function*
- *∇f is Lipschitz continuous with constant L*

- *$\mathcal{X} \subseteq \mathbb{R}^n$ be a closed and convex set*
- *$\mathcal{X}^*$ is nonempty.*

The gradient projection algorithm has the following properties:

1. *The inner loop searching for α_k will terminate in finite steps. And $\alpha_k = \bar{\alpha} > 0$ for all sufficiently large k.*
2. *$\{x_k\}$ converges to a **first order solution**.*
3. *The sequence $\{f(x_k)\}$ is monotonically decreasing. The decrease of objective function for all k given by*
$$f(x_{k+1}) \leq f(x_k) - (\frac{1}{\alpha} - \frac{1}{2\alpha_k})\|x_{k+1} - x_k\|_2^2.$$
4. $$f(x_{k+1}) - f_{opt} \leq \frac{\min_{x^* \in \mathcal{X}^*} \|x_0 - x^*\|_2^2}{2(k+1)\bar{\alpha}}$$

10.6.4.3 *Case study: sparsity regularization problem*

Definition 10.6.3 (target problem). *The target problem is*

$$\min_{x\in\mathbb{R}^n} f(x) + \lambda\|x\|_1,$$

where

- *$f : \mathbb{R}^n \to \mathbb{R}$ is **continuously differentiable convex function**.*
- *$\lambda > 0, \lambda \in \mathbb{R}$ is the regularization parameter.*

The proximal gradient iteration x_{k+1} is given by(component-wise):

$$[x_{k+1}]_i = \mathcal{S}(x_k - \alpha\nabla f(x_k), \alpha_k\lambda) \triangleq \begin{cases} [x_k - \alpha_k\nabla f(x_k)]_i - \alpha_k\lambda, \; if \; [x_k - \alpha_k\nabla f(x_k)]_i > \alpha_k\lambda \\ [x_k - \alpha_k\nabla f(x_k)]_i + \alpha_k\lambda, \; if \; [x_k - \alpha_k\nabla f(x_k)]_i < -\alpha_k\lambda \\ 0, otherwise \end{cases}.$$

That is, x_{k+1} is the solution to

$$\min_{x\in\mathbb{R}^n} f(x_k) + \nabla f(x_k)^T(x - x_k) + \lambda\|x\|_1 + \frac{1}{2\alpha_k}\|x - x_k\|_2^2$$

The algorithm is given below.

Algorithm 20: Iterative Shrinkage-Thresholding Algorithm with constant step size for L1 optimization

Input: Input initial $x_0 \in \mathbb{R}^n$ and Lipschitz constant L for ∇f

Set $k = 0$, and choose $\alpha \in (0, 1/L]$

repeat

  Evaluate $\nabla f(x_k)$.

  Compute $x_{k+1} = \mathcal{S}(x_k - \alpha \nabla f(x_k); \lambda\alpha)$.

  set $k = k + 1$.

until *termination condition*;

10.7 Notes on bibliography

For convex analysis, see [5]. For convex optimization algorithms, see [8].

For proximal algorithms, see [9].

For finance optimization, see Optimization Methods in Finance

A great online resource is the course page from Professor Stephen Boyd(http://stanford.edu/class/ee364b/resources.html). To learn about financial applications of convex optimization, see http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook_extra_exercises.pdf).

BIBLIOGRAPHY

1. Rockafellar, R. *Convex Analysis* ISBN: 9780691015866 (Princeton University Press, 1997).
2. Mordukhovich, B. S. & Nam, N. M. An easy path to convex analysis and applications. *Synthesis Lectures on Mathematics and Statistics* **6,** 1–218 (2013).
3. Bertsimas, D. & Tsitsiklis, J. N. *Introduction to linear optimization* (Athena Scientific Belmont, MA, 1997).
4. Bertsekas, D. *Nonlinear programming* ISBN: 9781886529007 (Athena Scientific, 2016).
5. Bertsekas, D. P. *Convex optimization theory* (Athena Scientific Belmont, 2009).
6. Robinson, D. *Convex optimization lecture notes* (Johns Hopkins University, 2015).
7. Nesterov, Y. *Introductory lectures on convex optimization: A basic course* (Springer Science & Business Media, 2013).
8. Bertsekas, D. P. & Scientific, A. *Convex optimization algorithms* (Athena Scientific Belmont, 2015).
9. Parikh, N., Boyd, S. P., *et al.* Proximal Algorithms. *Foundations and Trends in optimization* **1,** 127–239 (2014).