Convex Optimization Note

Garrick Lin

June 3, 2019

1 Affine Set

1.1 Definition

A set $\mathcal{A}$ is called affine set when it satisfied:

If $\mathbf{x_1} \in \mathcal{A}$ and $\mathbf{x_2} \in \mathcal{A}$, then, $\forall \theta \in \mathcal{R}$, $x = \theta \mathbf{x_1} + (1 - \theta)\mathbf{x_2}$ also belong to $\mathcal{A}$.

The expression $\theta \mathbf{x_1} + (1 - \theta)\mathbf{x_2}$ represent a line that cross through $\mathbf{x_1}$ and $\mathbf{x_2}$. Hence, affine set can be explained intuitively as:

Affine set is a set that contain the line which cross through any two point within this set.

1.2 Properties

Assume a affine set $\mathcal{A} \subseteq \mathcal{R}^n$. If $\mathcal{A}$ contain original point($\mathbf{0}$), then, $\mathcal{A}$ is a subspace. In order to prove this, $\mathcal{A}$ must satisfied following three rules:

1. $\mathcal{A}$ contain original point.

2. If $\mathbf{v} \in \mathcal{A}$, then, $\forall \theta \in \mathcal{R}$, $\theta \mathbf{v} \in \mathcal{A}$.

3. If $\mathbf{v_1}, \mathbf{v_2} \in \mathcal{A}$, then, $\mathbf{v_1} + \mathbf{v_2} \in \mathbf{A}$.

The first rule is already satisfied. For the second rule, $\forall \theta \in \mathcal{R}$, $\theta \mathbf{v}$ satisfied:

$$\theta \mathbf{v} = \theta \mathbf{v} + (1 - \theta)\mathbf{0}$$

Since $\mathcal{A}$ is affine and $\mathbf{v}, \mathbf{0} \in \mathcal{A}$, $\theta \mathbf{v} \in \mathcal{A}$ always true. For the third rule, there is:

$$\mathbf{v_1} + \mathbf{v_2} = 2(\frac{1}{2}\mathbf{v_1} + \frac{1}{2}\mathbf{v_2})$$

$\frac{1}{2}\mathbf{v_1} + \frac{1}{2}\mathbf{v_2}$ is in $\mathcal{A}$ as $\mathcal{A}$ is affine and $\frac{1}{2}\mathbf{v_1}, \frac{1}{2}\mathbf{v_2} \in \mathcal{A}$(According to the second rule, $\mathbf{v_1}, \mathbf{v_2} \in \mathcal{A}$, then, $\frac{1}{2}\mathbf{v_1}, \frac{1}{2}\mathbf{v_2} \in \mathcal{A}$). According to the second rule, $2(\frac{1}{2}\mathbf{v_1} + \frac{1}{2}\mathbf{v_2}) \in \mathcal{A}$

Therefore, any affine set $\mathcal{A}'$ can be thought as a subspace $\mathcal{V}$ with a transportation $\mathbf{v}$, which means, $\mathcal{A}' = \mathcal{V} + \mathbf{v}$. On the other hand, an affine set subtract some constant vector ($\mathcal{A}' - \mathbf{v}$), which make the new set contain original point, can form a subspace. Formally, for a affine set $\mathcal{A} = \{\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_n}\}$ and $\mathbf{a_i} \in \mathcal{A}$, the set:

$$\mathcal{A} - \mathbf{a_i} = \{\mathbf{a_1} - \mathbf{a_i}, \mathbf{a_2} - \mathbf{a_i}, \cdots, \mathbf{a_{i-1}}, \mathbf{0}, \mathbf{a_{i+1}}, \cdots, \mathbf{a_n} - \mathbf{a_i}\} \tag{1}$$

form a subspace. Furthermore, any affine set $\mathcal{A}$ has the form:

$$\mathcal{A} = \{\mathbf{x}|\, \mathcal{B}\mathbf{x} = \mathbf{c}\} \tag{2}$$

Proof: According to equation 1, $\mathcal{A} - \mathbf{a_i}$ is a subspace. Denote this subspace as $\mathcal{L} = \mathcal{A} - \mathbf{a_i}$. Assume the space that perpendicular to $\mathcal{L}$, denote as $\mathcal{L}^\perp$, has basis $\{\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n}\}$. Then, $\mathcal{L}$ is the set of vectors that perpendicular to the basis of $\mathcal{L}^\perp$. As a result, if we write the basis of $\mathcal{L}^\perp$ in matrix form, which is:

$$\mathcal{B} = \begin{bmatrix} \mathbf{b_1}^T \\ \mathbf{b_2}^T \\ \vdots \\ \mathbf{b_n}^T \end{bmatrix}$$

We have:
$$\mathcal{L} = \{\mathbf{y} | \, \mathcal{B}\mathbf{y} = 0\} \tag{3}$$

Equation 3 always hold as all vector in $\mathcal{L}^{\perp}$ is the linear combination of the basis, hence, if a vector $\mathbf{v}$ satisfied $\mathbf{b_1}^T\mathbf{v} = 0, \mathbf{b_2}^T\mathbf{v} = 0, \cdots, \mathbf{b_n}^T\mathbf{v} = 0$, then, $(\theta_1\mathbf{b_1} + \theta_2\mathbf{b_2} + \cdots \theta_n\mathbf{b_n})^T\mathbf{v} = 0$, which means,$\mathbf{v}$ is perpendicular to any vector in $\mathcal{L}^{\perp}$ if $\mathbf{v}$ is perpendicular to the basis of $\mathcal{L}^{\perp}$.

Since the vector in $\mathcal{L}$ come from $\mathcal{A}$ subtract $\mathbf{a_i}$, for all $\mathbf{y} \in \mathcal{L}$, there must exists $\mathbf{x} \in \mathcal{A}$ such that:

$$\mathbf{y} = \mathbf{x} - \mathbf{a_i} \tag{4}$$

By combining equation 4 and equation 3, we have:

$$\mathcal{A} = \{\mathbf{x} | \, \mathcal{B}(\mathbf{x} - \mathbf{a_i}) = 0\}$$

Which is equivalent to :

$$\mathcal{A} = \{\mathbf{x} | \, \mathcal{B}\mathbf{x} = \mathcal{B}\mathbf{a_i}\}$$

and has the same form as equation 2, where $\mathbf{c} = \mathcal{B}\mathbf{a_i}$.

2 Hyper-plane

2.1 Definition

In a $\mathcal{R}^n$ space, the hyperplane is defined as a $n-1$ dimension subspace plus a translate $\mathbf{x_0}$, which is perpendicular to a one dimension vector $\mathbf{w}$. The formal mathematical expression of hyper-plane $\mathcal{H}$ is:

$$\mathcal{H} = \{\mathbf{y} + \mathbf{x_0} | \, \mathbf{w}^T\mathbf{y} = 0\} \tag{5}$$

We can denote $\mathbf{x} = \mathbf{y} + \mathbf{x_0}$, hence, equation 5 become:

$$\mathcal{H} = \{\mathbf{x} | \, \mathbf{w}^T(\mathbf{x} - \mathbf{x_0}) = 0\} \tag{6}$$

It is clearly that when $\mathbf{x} = \mathbf{x_0}$, $\mathbf{w}^T(\mathbf{x_0} - \mathbf{x_0}) = 0$ always hold, hence, $\mathbf{x_0}$ is a point within hyper-plane. As a result, equation 6 can be explained intuitively as: Hyper-plane is a set of any point $\mathbf{x}$ which satisfied the vector $\mathbf{x} - \mathbf{x_0}$(start from a fixed point $\mathbf{x_0}$ which belong to this hyper-plane, end at $\mathbf{x}$) is perpendicular to a normal vector $\mathbf{w}$.

Equation 6 give us another way to represent hyper-plane:

$$\mathcal{H} = \{\mathbf{x} | \, \mathbf{w}^T\mathbf{x} = \mathbf{b}\} \tag{7}$$

Where $\mathbf{b} = \mathbf{w}^T\mathbf{x_0}$.

2.2 Half-space

Intuitively, a hyper-plane split the whole space into two part, each of them is called half-space. The formal mathematical expression for half-space, base on equation 6, is:

$$\{\mathbf{x} | \, \mathbf{w}^T(\mathbf{x} - \mathbf{x_0}) < 0\}$$

or

$$\{\mathbf{x} | \, \mathbf{w}^T(\mathbf{x} - \mathbf{x_0}) > 0\}$$

Which corresponding to the two split space. In geometric, two vector $\mathbf{v_1}$ and $\mathbf{v_2}$ satisfied: $\mathbf{v_1}^T\mathbf{v_2} > 0$ if their angle smaller than 90 degree, otherwise, $\mathbf{v_1}^T\mathbf{v_2} < 0$. As a result, the half-space $\{\mathbf{x} | \, \mathbf{w}^T(\mathbf{x} - \mathbf{x_0}) < 0\}$ is the set of point $\mathbf{x}$ which satisfied the vector $\mathbf{x} - \mathbf{x_0}$ have a angle greater than 90 degree with a normal vector $\mathbf{w}$. The half-space $\{\mathbf{x} | \, \mathbf{w}^T(\mathbf{x} - \mathbf{x_0}) > 0\}$ can be explained in the same way except the angle is smaller than 90 degree.

Furthermore, according to 7, the expression of half-space can also be written as:

$$\{\mathbf{x} | \, \mathbf{w}^T\mathbf{x} < \mathbf{b}\}$$

Where $\mathbf{b} = \mathbf{w}^T\mathbf{x_0}$.

3 Cone

3.1 Definition

A cone $\mathcal{K}$ is the set of any point $\mathbf{x}$ that satisfied: If $\mathbf{x} \in \mathcal{K}$, then, for any $\theta \geq 0$, $\theta \mathbf{x} \in \mathcal{K}$. A **convex** cone, as its' name suggest, is a cone which is also a convex set.

3.2 Property of Convex Cone

Assume a cone $\mathcal{K}$ is a convex cone, then, it satisfied:

If $\mathbf{x_1} \in \mathcal{K}$ and $\mathbf{x_2} \in \mathcal{K}$, then, for any $0 \leq \theta \leq 1$, $\theta \mathbf{x_1} + (1 - \theta) \mathbf{x_2} \in \mathcal{K}$.

Furthermore, for any $p, q > 0$, the expression $p \mathbf{x_1} + q \mathbf{x_2}$ can be written as:

$$p \mathbf{x_1} + q \mathbf{x_2} = (p + q)(\frac{p}{p + q} \mathbf{x_1} + \frac{q}{p + q} \mathbf{x_2}) \tag{8}$$

Following the property of convex set, the point $(\frac{p}{p+q} \mathbf{x_1} + \frac{q}{p+q} \mathbf{x_2}) \in \mathcal{K}$. Additionally, since $(p + q) \in \mathcal{R}$ and $p + q > 0$, the definition of cone indicate $(p + q)(\frac{p}{p+q} \mathbf{x_1} + \frac{q}{p+q} \mathbf{x_2}) \in \mathcal{K}$. As a result, it can be seen within a convex cone $\mathcal{K}$:

If $\mathbf{x_1}, \mathbf{x_2} \in \mathcal{K}$, then, for any $\theta_1, \theta_2 > 0$, $\theta_1 \mathbf{x_1} + \theta_2 \mathbf{x_2} \in \mathcal{K}$

3.3 Dual Cone

In geometric, a point $\mathbf{x}$ can be defined as the end point of a vector that start with the original point($\mathbf{0}$) and terminate at point $\mathbf{x}$, which means, any point $\mathbf{x}$ can also be thought as a vector $\mathbf{x} - \mathbf{0}$. Base on this, the dual cone of a cone $\mathcal{K}$, denote as $\mathcal{K}^*$, is defined as the set of any point λ which satisfied $\lambda^T \mathbf{x} > 0$ for all $\mathbf{x} \in \mathcal{K}$, which means, the **vector** λ have the angle smaller than 90 degree for any **vector** $\mathbf{x} - \mathbf{0}$ defined by the point $\mathbf{x}$ in $\mathcal{K}$.

Furthermore, the expression $\lambda^T \mathbf{x} > 0$ define a half-space according to equation 6, where the fixed point $\mathbf{x_0}$ is the original point. As a result, the dual cone of $\mathcal{K}$ can also be illustrated as:

A point λ is in $\mathcal{K}^*$ if the half-space with the **vector** λ as normal vector, which expressed as $\lambda \mathbf{x} > 0$, contain the original cone $\mathcal{K}$.

3.4 Generalized Inequality

We can define the generalized inequality based on the concept of cone:

$\mathbf{x}$ is smaller than $\mathbf{y}$ respect to a specific cone $\mathcal{K}$ if $\mathbf{y} - \mathbf{x} \in \mathcal{K}$. Denote as: $\mathbf{x} \prec_{\mathcal{K}} \mathbf{y}$

Start from the definition of generalized inequality, we can define **minimum** and **minimal** value of a set $\mathcal{S}$ with respect to a specific cone $\mathcal{K}$:

The point $\mathbf{x_0} \in \mathcal{S}$ is said to be the **minimum** of $\mathcal{S}$ if for any other $\mathbf{y} \in \mathcal{S}$, the expression $\mathbf{x} \preceq_{\mathcal{K}} \mathbf{y}$ always holds.

Before we move to the definition of **minimal** point, we have to realize an important fact with respect to generalized inequality: When $\mathbf{y} \not\preceq_{\mathcal{K}} \mathbf{x}$ hold, it **does not** mean $\mathbf{x} \preceq_{\mathcal{K}} \mathbf{y}$, which means, even though $\mathbf{y}$ is not smaller than $\mathbf{x}$ (with respect to a specific cone $\mathcal{K}$), it **does not** implies that $\mathbf{x}$ is smaller or equal to $\mathbf{y}$. This fact implies that a point smaller than any other element of a set is not equal to there is no other element within this set that smaller than the point. We can now introduce the definition of **minimal** point with respect to cone $\mathcal{K}$:

A point $\mathbf{x_0}$ is minimal point of $\mathcal{S}$ if for any $\mathbf{y} \in \mathcal{S}$, $\mathbf{y} \not\preceq_{\mathcal{K}} \mathbf{x_0}$ always hold unless $\mathbf{y} = \mathbf{x_0}$

Which means, if $\mathbf{x_0}$ is a minimal point of $\mathcal{S}$, there exist no element $\mathbf{y}$ within $\mathcal{S}$ which satisfied $\mathbf{y} \preceq_{\mathcal{K}} \mathbf{x_0}$(There is no element in $\mathcal{S}$ that smaller than $\mathbf{x_0}$). Furthermore, according to the fact that mentioned above, for some $\mathbf{y} \in \mathcal{S}$, even though $\mathbf{y} \not\preceq_{\mathcal{K}} \mathbf{x_0}$, $\mathbf{x_0}$ does **not** need to satisfy $\mathbf{x_0} \preceq_{\mathcal{K}} \mathbf{y}$. As a result, $\mathbf{x_0}$ does not need to be the minimum point.

3.5 Minimum and Minimal with Dual Cone Perspective

If a point $\mathbf{x_0}$ is a minimum point of $\mathcal{S}$ with respect to cone $\mathcal{K}$, then, for any $\mathbf{x} \in \mathcal{S}$, there is:

$$\mathbf{x} - \mathbf{x_0} \in \mathcal{K}$$

According to the definition of dual cone, for the element $\mathbf{v}$ within $\mathcal{K}$, it must satisfy:

$$\lambda^T \mathbf{b} > 0$$

Where $\lambda \in \mathcal{K}^*$. Since $\mathbf{x} - \mathbf{x_0} \in \mathcal{K}$, we have:

$$\lambda^T(\mathbf{x} - \mathbf{x_0}) > 0$$

Which define a hyper-plane and a half-space with $\mathbf{x_0}$ as the fixed point according to equation 6. As a result, the minimum point of a set $\mathcal{S}$ with respect to a cone $\mathcal{K}$ and its' dual cone $\mathcal{K}^*$ can be illustrated as:

> If $\mathbf{x_0}$ is minimum point, then, for any $\lambda \in \mathcal{K}^*$, the hyper-plane and half-space defined by λ and $\mathbf{x_0}$: $\lambda^T(\mathbf{x} - \mathbf{x_0})$ should contain entire $\mathcal{S}$.

4 Convex Function

4.1 Definition

A function $f(\mathbf{x})$ is called convex function if it satisfied:

$$f(\theta \mathbf{x_2} + (1 - \theta)\mathbf{x_2}]) \leq \theta \mathbf{x_1} + (1 - \theta)\mathbf{x_2} \tag{9}$$

for any $\mathbf{x_1}, \mathbf{x_2} \in \mathbf{dom} f$ and $0 \leq \theta \leq 1$.

4.2 Conditions

A function $f(\mathbf{x})$ is a convex function if it satisfied one of the following conditions:

4.2.1 First-Order Condition

If $f(\mathbf{x})$ is differentiable, then, for any $\mathbf{x} \in \mathbf{dom} f$ and an arbitrary $\mathbf{x_0}$, the first-order derivative if f should satisfied:

$$f(\mathbf{x}) \geq f(\mathbf{x_0}) + \nabla f(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0}) \tag{10}$$

The right hand side of this inequality is the first-order Taylor expression, which can be written as:

$$g(\mathbf{x}) = f(\mathbf{x_0}) + \nabla f(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0})$$

Therefore, the first-order condition can be explained as: For any point $\mathbf{x}$, the value of function f at this point is greater or equal to the value of g at the same point $\mathbf{x}$. In $\mathcal{R}^2$, the function $g(\mathbf{x})$ define a line that is tangent to $f(\mathbf{x})$ at $(\mathbf{x_0}, f(\mathbf{x_0}))$. Moreover, in $\mathcal{R}^n$, $g(\mathbf{x})$ indeed define a *support* hyper-plane as the function $g(\mathbf{x})$ can be written as:

$$\nabla f(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0}) - (g(\mathbf{x}) - f(\mathbf{x_0})) = 0$$

Which is the same as:

$$\begin{bmatrix} \nabla f(\mathbf{x_0}) \\ -1 \end{bmatrix}^T (\begin{bmatrix} \mathbf{x} \\ g(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \mathbf{x_0} \\ f(\mathbf{x_0}) \end{bmatrix}) = 0 \tag{11}$$

It is obvious that the above equation define a hyper-plane. Hence, the first-order condition can also be expressed as: For any $\mathbf{x} \in \mathbf{dom} f$, the point $(\mathbf{x}, f(\mathbf{x}))$ is above the hyper-plane defined by equation 11 through some fixed point $(\mathbf{x_0}, \mathbf{f}(\mathbf{x_0}))$, which means, it is in the half-space given by this hyper-plane. This can be proved through:

$$\begin{bmatrix} \nabla f(\mathbf{x_0}) \\ -1 \end{bmatrix}^T (\begin{bmatrix} \mathbf{x} \\ f(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} \mathbf{x_0} \\ f(\mathbf{x_0}) \end{bmatrix}) \leq 0$$

Which define a half-space. Furthermore, when $\nabla f(\mathbf{x_0}) = 0$ in some point $(\mathbf{x_0}, f(\mathbf{x_0}))$, the equation 10 become:

$$f(\mathbf{x}) \geq f(\mathbf{x_0})$$

Hence, $f(\mathbf{x_0})$ is the minimum value of function f, which is why for convex function, the point that first-order derivative equal to zero is the minimizer of this function.

Indeed, we can have a geometry interpretation of term $\nabla f(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0})$ over $\mathcal{R}^2$:
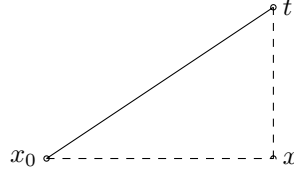
The straight line that pass through has the slope $\nabla f(\mathbf{x_0})$, the term $\nabla f(x_0)(x - x_0)$ is actually the length of the line segment between points t and x. Hence, if $\nabla f(x_0)(x - x_0) \geq 0$, the length of line segment between x and t is positive and hence the point t is above the point x_0. By contrast, if $\nabla f(x_0)(x - x_0) \leq 0$, the point t is below the point x_0. Furthermore, we can also see that the term $f(x_0) + \nabla f(x_0)(x - x_0)$ is actually the value of $f(x_0)$ plus the length of line segment between t and x.

4.2.2 Second-Order Condition

If function $f(\mathbf{x})$ is twice differentiable, then, $f(\mathbf{x})$ is convex if and only if it's second-order Hessian Matrix is positive semi-definite:

$$\nabla^2 f(\mathbf{x}) \succeq 0 \tag{12}$$

This condition is more often be used to verify whether a function is convex than the first-order condition as it is more convenient to compute Hessian Matrix and determine whether it is positive semi-definite. By contrast, the first-order condition works more like a property that indicate for convex function, it's minimum value exist at point $\mathbf{x_0}$ which satisfied $\nabla f(\mathbf{x_0}) = 0$.

Even though the second-order condition provide a good way to verify the convexity of a function f, a more easier way to verify the convexity is to prove that $f(\mathbf{x})$ is constructed by some operations that preserve convexity. These operation will be describe in the next part.

4.3 Operation that Preserve Convexity

A function $f(\mathbf{x})$ is convex if it can be constructed through one of the following operation:

4.3.1 Non-negative Weighted Sum

Function $f(\mathbf{x})$ is convex if it can be constructed through:

$$f(\mathbf{x}) = \sum_{i=1}^{n} a_i f_i(\mathbf{x})$$

Where $a_i \geq 0$ and $f_i(\mathbf{x})$ is convex.

4.3.2 Point-wise maximum

Function $f(\mathbf{x})$ is convex if it is the point-wise maximum of some convex function:

$$f(\mathbf{x}) = max\{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_n(\mathbf{x})\}$$

Where $f_i(\mathbf{x})$ is convex. Based on this property, it can also be proved that function $f(\mathbf{x})$ is convex if it is the supreme of function $g(\mathbf{x}, \mathbf{y})$ over $\mathbf{y}$:

$$f(\mathbf{x}) = \sup_{\mathbf{y}} g(\mathbf{x}, \mathbf{y})$$

As $f(\mathbf{x})$ can be constructed through:

$$f(\mathbf{x}) = max\{g(\mathbf{x}, \mathbf{y_1}), g(\mathbf{x}, \mathbf{y_2}), \cdots, g(\mathbf{x}, \mathbf{y_n})\}$$

Where $\mathbf{dom}_y g(\mathbf{x}, \mathbf{y}) = \{\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_n}\}$

4.3.3 Composition

Function $f(\mathbf{x})$ is convex if it is a composition of other function:

$$f(\mathbf{x}) = h(g(\mathbf{x}))$$

with function $h(\mathbf{y}), g(\mathbf{y})$ under some constrains, which can be derived directly from:

$$\nabla_2 f(\mathbf{x}) = \nabla_2 h(g(\mathbf{x}))(\nabla g(\mathbf{x}))^2 + \nabla h(g(\mathbf{x}))\nabla_2 g(\mathbf{x}) \geq 0 \tag{13}$$

e.g. If $g(\mathbf{y}), h(\mathbf{y})$ is convex and $h(\mathbf{y})$ is non-decreasing, then, $\nabla_2 h(g(\mathbf{x})) \geq 0$, $\nabla h(g(\mathbf{x})) \geq 0$ and $\nabla_2 g(\mathbf{x}) \geq 0$. Hence, equation 13 always hold, which means $f(\mathbf{x})$ is convex. The other constrains can be derived based on the similarly.

4.4 Epigraph

The epigraph is another thing that we can used to prove convexity. Geometrically, the epigraph of a function $f(\mathbf{x})$ is the area that above the graph of $f(\mathbf{x})$. Mathematically, the epigraph of a function $f(\mathbf{x})$ can be defined as a set of points that satisfied:

$$\{(\mathbf{x}, t) | f(\mathbf{x}) \leq t\}$$

The function $f(\mathbf{x})$ is convex if the point within its' epigraph form a convex set.

5 Conjugate Function

5.1 Definition

The conjugate function of $f(\mathbf{x})$, denote as f^*, can be defined as:

$$f^*(\mathbf{y}) = \sup_{\mathbf{x}} \left(\mathbf{y}^T \mathbf{x} - f(\mathbf{x}) \right) \tag{14}$$

In order to complete the definition of conjugate function $f^*(\mathbf{y})$, we need to define the domain of f^*, *i.e.* the range of $\mathbf{y}$: The domain of f^* is the $\mathbf{y}$ that make $f^*(\mathbf{y})$ finite:

$$\mathbf{dom} f^*(\mathbf{y}) = \{\mathbf{y} | -\infty < f^*(\mathbf{y}) < \infty\}$$

5.2 Find Conjugate Function

A simple procedure that may be used to find conjugate function of $f(\mathbf{x})$ is:

1. Establish the domain $\mathbf{dom} f^*(\mathbf{y})$, this may be done by observing or experiment, *e.g.* start from lower dimension (2-dimension), find the range of $\mathbf{y}$ that cause $f^*(\mathbf{y})$ goes to infinity and exclude these range.

2. Find the supreme of $\mathbf{y}^T \mathbf{x} - f(\mathbf{x})$ over $\mathbf{x}$. Some general method to do this are:

   - Take the gradient of $\mathbf{x}$ and set it to zero.
   - By observing.

6 Quasiconvex Function

Before we introduce the definition of quasiconvex function, we need first familiar with the concept of α-sublevel set. The α-sublevel set $\mathcal{S}_\alpha$ of a function $f(\mathbf{x})$ is the set of $\mathbf{x}$ that satisfied $f(\mathbf{x}) \leq \alpha$:

$$\mathcal{S}_\alpha = \{\mathbf{x} | f(\mathbf{x}) \leq \alpha\} \tag{15}$$

Based on the concept of α-sublevel set, we can give the definition of quasiconvex function:

The function $f(\mathbf{x})$ is quasiconvex if and only if for any $\alpha \in \mathcal{R}$, the α-sublevel set of $f(\mathbf{x})$ is convex.

Indeed, for any $\mathbf{y} \in \mathbf{dom} f$, we can make $\alpha = f(\mathbf{y})$. Hence, the α-sublevel set $\mathcal{S}_\alpha$ is convex. By noticing that $f(\mathbf{y}) \leq f(\mathbf{y})$ always hold, we can conclude that $\mathbf{y} \in \mathcal{S}_\alpha$. Therefore, for any $\mathbf{x} \in \mathcal{S}_\alpha$ and $\mathbf{y}$, $\theta \mathbf{x} + (1-\theta)\mathbf{y} \in \mathcal{S}_\alpha$ and hence, $f(\theta \mathbf{x} + (1-\theta)\mathbf{y}) \leq f(\mathbf{y})$. As a result, we can draw a simple conclude about quasiconvex function f: For any $\mathbf{x}, \mathbf{y} \in \mathbf{dom} f$ and $0 \leq \theta \leq 1$, there is

$$f(\theta \mathbf{x} + (1-\theta)\mathbf{y}) \leq max\{f(\mathbf{x}), f(\mathbf{y})\} \tag{16}$$

This is the first order property of quasiconvex function.

7 Convex Optimization Problem

After we have described some basic concept, we can now start to formalize the concept of convex optimization problem. We will first give the definition of general optimization problem.

7.1 General Optimization Problem

The general optimization problem has the stand form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ s.t. \quad & f_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \cdots n \\ & h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots m \end{aligned} \tag{17}$$

The *domain* $\mathcal{D}$ of this optimization problem is:

$$\mathcal{D} = \mathbf{dom} f_0 \cap \cdots \cap \mathbf{dom} f_n \cap \mathbf{dom} h_1 \cap \cdots \cap \mathbf{dom} h_m$$

Indeed, we can think of that the constrains $f_i(\mathbf{x})$ and $h_i(\mathbf{x})$ define a set $\mathcal{C}$ of $\mathbf{x}$:

$$\mathcal{C} = \{\mathbf{x} |\, f_i(\mathbf{x}) \leq 0 (i = 1, \cdots n), h_j(\mathbf{x}) = 0 (j = 1, \cdots m)\}$$

And the goal of optimization problem is to find a $\mathbf{x}^* \in \mathcal{C}$ that minimize $f_0(\mathbf{x})$. The set $\mathcal{C}$ here is called feasible set and a $\mathbf{x}$ is called feasible if $\mathbf{x} \in \mathcal{C}$. Generally, we are more interesting in a special type of optimization problem, which is convex optimization problem.

7.2 Convex Optimization Problem

Convex optimization problem is a specific type of optimization problem, in which, the function $f_i(\mathbf{x})\,(i = 0, 1, \cdots n)$ is convex and $h_j(\mathbf{x})\,(j = 1, 2, \cdots m)$ is affine(linear). Mathematically, convex optimization problem can be expressed as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ s.t. \quad & f_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \cdots n \\ & \mathcal{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{18}$$

Where $f_i(\mathbf{x})\,(i = 0, 1, \cdots n)$ is convex.

7.3 Special Cases

Some convex optimization problem with special constrain have typical solutions. Particularly, the convex optimization problem that only have equality constrains:

$$\begin{aligned} \min_{x} \quad & f_0(\mathbf{x}) \\ s.t. \quad & \mathcal{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

In this case, we first notice that if we can find a $\mathbf{x_0}$ that satisfied $\mathcal{A}\mathbf{x_0} = \mathbf{b}$, then, the following equation:

$$\mathcal{A}(\mathbf{z} + \mathbf{x_0}) = \mathbf{b} \tag{19}$$

always hold for any $\mathbf{z}$ that satisfied $\mathcal{A}\mathbf{z} = 0$. Hence, we can conclude that for every $\mathbf{x}$ that satisfied $\mathcal{A}\mathbf{x} = \mathbf{b}$, it can be decomposed as a variable $\mathbf{z}$ plus a fixed vector $\mathbf{x_0}$:

$$\mathbf{x} = \mathbf{z} + \mathbf{x_0} \tag{20}$$

Where $\mathbf{z}$ is in the null space of $\mathcal{A}$ ($\mathbf{z} \in \mathcal{N}(\mathcal{A})$). Furthermore, we can find the biases of null space of $\mathcal{A}$, denote as $\mathcal{M}$ and express $\mathbf{z}$ as:

$$\mathbf{z} = \mathcal{M}\mathbf{y} \tag{21}$$

Where $\mathbf{y} \in \mathcal{R}^n$. The Equation 21 means that $\mathbf{z}$ is arbitrary linear combination of the biases of $\mathcal{N}(\mathcal{A})$. By the combination of Equation 20 and 21, the original problem can be refined as:

$$\min_{\mathbf{y}} \quad f_0(\mathcal{M}\mathbf{y} + \mathbf{x_0})$$

With the feasible set $\mathcal{C}_{\mathbf{y}} = \{\mathbf{y} |\, \mathbf{y} \in \mathcal{R}^n\}$. By transforming the feasible to $\mathcal{R}^n$, we reduce the difficulty to find the optimal value of $\mathbf{f_0}$. Indeed, this is the idea that where *Lagrangian* come from, which will be explained in detail later.

7.4 Equivalent Problem

By inspecting the transforming we have done in section 7.3, it can be seen that even though we introduce a new variable $\mathbf{y}$ and define a new feasible set $\mathcal{C}_{\mathbf{y}}$, we did not change the original feasible set $\mathcal{C}_{\mathbf{x}}$ of $\mathbf{x}$, *i.e.* $\mathcal{C}_{\mathbf{x}} = \{\mathbf{x} \mid \mathcal{A}\mathbf{x} = \mathbf{b}\} = \{\mathbf{x} \mid \mathbf{x} = \mathcal{M}\mathbf{y} + \mathbf{x_0}, \mathbf{y} \in \mathcal{R}^n\}$. Hence, we did not change the optimal value of f_0.

This observation tell us that we can construct a equivalent optimization problem if we can guarantee the new optimization problem has the same optimal value with the original problem (In some cases, the feasible set may not even need to be equal). One particular example is the convex optimization problem with objective function $f_0(\mathbf{x})$ satisfied $f_0(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})}$ and $p(t\mathbf{x}) = tp(\mathbf{x}), q(t\mathbf{x}) = tq(\mathbf{x})$ for any t:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})}$$
$$s.t. \quad f_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \cdots n$$
$$h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots m$$

Since $q(t\mathbf{x}) = tq(\mathbf{x})$, for any $\mathbf{x}$ and $q(\mathbf{x})$, we can fetch out a multiplier t from $\mathbf{x}$:

$$q(\mathbf{x}) = q(t\mathbf{z})$$
$$= tq(\mathbf{z})$$

and force $q(\mathbf{z}) = 1$. This is equivalent to introduce a new variable $\mathbf{z}$ that satisfied $\mathbf{x} = t\mathbf{z}$. In the meantime, the same operation should apply to $p(\mathbf{x})$, hence, $p(\mathbf{x}) = tp(\mathbf{z})$. As a result, the objective function become:

$$f_0(\mathbf{x}) = \frac{tp(\mathbf{z})}{tq(\mathbf{z})}$$
$$= \frac{p(\mathbf{z})}{q(\mathbf{z})} \tag{22}$$
$$= p(\mathbf{z})$$

This equation means that for any feasible $\mathbf{x}$, we can find a corresponding $\mathbf{z}$ that satisfied $q(\mathbf{z}) = 1$ and $f_0(\mathbf{x}) = p(\mathbf{z})$. As a result, we can focus on a subset of the original feasible set $\mathcal{C}$ that satisfied $\{\mathbf{z} \mid \mathbf{z} \in \mathcal{C} \ and \ q(\mathbf{z}) = 1\}$ and construct a equivalent optimization problem:

$$\min_{\mathbf{z}} \quad p(\mathbf{z})$$
$$s.t. \quad f_i(\mathbf{z}) \leq 0 \quad i = 1, 2, \cdots n$$
$$h_j(\mathbf{z}) = 0 \quad j = 1, 2, \cdots m$$
$$q(\mathbf{z}) - 1 = 0$$

We can verify these two problem is equivalent directly from Equation 22 which says these two problems have the same optimal value. More general, we can multiple both $p(\mathbf{x})$ and $q(\mathbf{x})$ with a variable t and force $tq(\mathbf{x}) = 1$. As a result, we successfully remove the numerator $q(\mathbf{x})$ and construct an equivalent problem as follow:

$$\min_{\mathbf{x}} \quad tp(\mathbf{x})$$
$$s.t. \quad f_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \cdots n$$
$$h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots m$$
$$tq(\mathbf{x}) - 1 = 0$$

After we have gave the formal definition of convex optimization problem, we can start to introduce the method used to solve convex optimization problem. Indeed, most optimization problem, include convex optimization problem, are hard to solve. Hence, instead of solving the problem directly, we may consider to find the lower bound d^* that satisfied $d^* \leq p^*$ (p^* is the optimal value of original problem), which can be done through constructing the Lagrangian of original problem.

8 Lagrangian

For a convex optimization problem:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x})$$
$$s.t. \quad f_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \cdots n$$
$$h_j(\mathbf{x}) \leq 0 \quad j = 1, 2, \cdots m$$

with domain $\mathcal{D}$ and optimal value $p^* = f_0(\mathbf{x}^*)$. In order to find the lower bound of p^*, we can start with finding the lower bound of $f_0(\mathbf{x})$ for any feasible $\mathbf{x}$. Indeed, for any $\mathbf{x} \in \mathcal{C}$, it satisfied $f_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$. Hence, we can find $\lambda_i \geq 0$ and $\nu_j \in \mathcal{R}$ to construct the inequality:

$$f_0(\mathbf{x}) \geq f_0(\mathbf{x}) + \underbrace{\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x})}_{\text{negative}} + \underbrace{\sum_{j=1}^{m} \nu_j h_j(\mathbf{x})}_{\text{zero}} \tag{23}$$

Additionally, for any feasible $\mathbf{x}$, the following inequality always hold:

$$f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \geq \inf_{\mathbf{x} \in \mathcal{C}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \tag{24}$$

Combine Equation 23 and 24, we have:

$$f_0(\mathbf{x}) \geq \inf_{\mathbf{x} \in \mathcal{C}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \tag{25}$$

Technical speaking, the operation over feasible set may be difficult. Hence, we prefer to consider the problem over the whole domain $\mathcal{D}$. Firstly, we notice that the feasible set $\mathcal{C}$ is a subset of $\mathcal{D}$, hence, we have:

$$\inf_{\mathbf{x} \in \mathcal{C}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \geq \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \tag{26}$$

as the infimum of a set is always small or equal to the infimum of it's subset. Combine Equation 25 and 26, we have:

$$f_0(\mathbf{x}) \geq \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \tag{27}$$

Particularly, we can let $\mathbf{x} = \mathbf{x}^*$ and get:

$$f_0(\mathbf{x}^*) = p^* \geq \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \tag{28}$$

We can denote:

$$\mathcal{L}(\mathbf{x}, \lambda, \nu) = f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \quad (\lambda_i \geq 0) \tag{29}$$

which is the *Lagrangian* of the original optimization problem, and

$$\begin{aligned} g(\lambda, \nu) &= \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j h_j(\mathbf{x}) \right) \\ &= \inf_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \lambda, \nu) \end{aligned} \tag{30}$$

which is called *Lagrange Dual Function*. The Equation 28 construct a lower bound of p^*, which is $g(\lambda, \nu)$. Particularly, each pair of (λ, ν) corresponding to one lower bound. Hence, we may want to find a best lower bound that has the smallest gap with p^*. It is obvious that this best lower bound is the maximum of $g(\lambda, \nu)$ and hence, we formalize a new optimization problem:

$$\begin{aligned} \max_{\lambda, \nu} \quad & g(\lambda, \nu) \\ s.t. \quad & \lambda \succeq 0 \end{aligned} \tag{31}$$

The optimal value of Problem 31, denote as d^*, is the best lower bound of p^* and the Problem 31 is called *Lagrange Dual Problem*.

Hence, we can conclude a simple method that solve the convex optimization problem:

1. **Finding the Lagrange Dual Function**: This step can usually be done through taking the first-order derivative of $\mathbf{x}$ and making it equal to zero as the domain we consider is the whole set $\mathcal{D}$.

2. **Solve the Langrange Dual Problem**: This step can give us a best lower bound of p^*. Particularly, when the problem is convex, this lower bound has zero-gap with p^*.

8.1   Geometry Interpretation

For any optimization with optimal value $p^* = f_0(\mathbf{x}^*)$, we can let $\mathbf{u} = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}$, $\mathbf{w} = \begin{bmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ \vdots \\ h_m(\mathbf{x}) \end{bmatrix}$ and $t = f_0(\mathbf{x})$.

Therefore, the point $(\mathbf{u}, \mathbf{w}, t)$ form a set $\mathcal{G}$. Furthermore, the Lagrangian under this situation is:

$$\mathcal{L} = \begin{bmatrix} \lambda^T & \nu^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ t \end{bmatrix}$$

According to Equation 30, we have:

$$\begin{bmatrix} \lambda^T & \nu^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ t \end{bmatrix} \geq g(\lambda, \nu)$$

Which define a half-space that contain the set $\mathcal{G}$. By inspecting the hyper-plane $\begin{bmatrix} \lambda^T & \nu^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \\ t \end{bmatrix} = g(\lambda, \nu)$, it is easy to verify that the point $(0, 0, g(\lambda, \nu))$ lies on this hyper-plane. Hence, the intersection point of the hyper-plane and t-axis has the t-component value equal to $g(\lambda, \nu)$.

Furthermore, when the set $\mathcal{G}$ is convex, *i.e.* the original problem is convex optimization problem, there exist a pair (λ^*, ν^*) that satisfied $g(\lambda^*, \nu^*) = p^*$. Indeed, we can construct a set $\mathcal{B} = \{(0, 0, t) | t < p^*\}$. It is obvious that $\mathcal{B}$ is convex and has no intersection with $\mathcal{G}$ as all the points within $\mathcal{G}$ has the t-component value greater or equal to p^*. Consider both $\mathcal{G}$ and $\mathcal{B}$ are convex and do not intersect, there exist a hyper-plane defined by $\begin{bmatrix} \lambda \\ \nu \\ \sigma \end{bmatrix}$ and α that separate these two sets:

$$\begin{cases} \lambda^T \mathbf{u} + \nu^T \mathbf{w} + \sigma t \geq \alpha \\ \lambda^T 0 + \nu^T 0 + \sigma q < \alpha \end{cases} \tag{32}$$

Where $(\mathbf{u}, \mathbf{w}, t) \in \mathcal{G}$ and $(0, 0, q) \in \alpha$. Particularly, since $q < p^*$, we have:

$$\begin{cases} \lambda^T \mathbf{u} + \nu^T \mathbf{w} + \sigma t \geq \alpha \\ \sigma p^* \leq \alpha \end{cases} \tag{33}$$

which implies:

$$\lambda^T \mathbf{u} + \nu^T \mathbf{w} + \sigma t \geq \alpha \geq \sigma p^* \tag{34}$$

We can divide σ on both sides and get:

$$\frac{\lambda^T}{\sigma} \mathbf{u} + \frac{\nu}{\sigma} \mathbf{w} + t \geq p^* \tag{35}$$

If we let $\lambda^* = \frac{\lambda}{\sigma}$ and $\nu^* = \frac{\nu}{\sigma}$. The Equation 35 actually define a half space with vector $\begin{bmatrix} \lambda^* \\ \nu^* \\ 1 \end{bmatrix}$ and $g(\lambda^*, \nu^*) = p^*$.

8.2   Strong and Weak Duality

According to above discussion, for any $\lambda \succeq 0$ and $\nu \in \mathcal{R}^m$, there is: $g(\lambda, \nu) \leq p^*$. This is called weak duality. Particularly, in some cases, there exist $\lambda^* \succeq 0$ and ν^* that satisfied $g(\lambda^*, \nu^*) = \max\limits_{\lambda, \nu} g(\lambda, \nu)$ and $g(\lambda^*, \nu^*) = p^*$, which means, there is no gap between the best lower bound and the optimal value p^*. This is called strong duality.

8.3 KKT Condition

For a optimization problem with optimal value $p^* = f_0(\mathbf{x}^*)$. If the strong duality hold, which means, there exist $\lambda^* \geq 0$ and ν^* that satisfied $g(\lambda^*, \nu^*) = p^*$, then, we have:

$$
\begin{aligned}
f_0(\mathbf{x}^*) &= g(\lambda^*, \nu^*) \\
&= \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{n} \lambda_i^* f_i(\mathbf{x}) + \sum_{j=1}^{m} \nu_j^* h_j(\mathbf{x}) \right) \\
&\leq f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \lambda_i^* f_i(\mathbf{x}^*) + \sum_{j=1}^{m} \nu_j^* h_j(\mathbf{x}^*) \\
&\leq f_0(\mathbf{x}^*)
\end{aligned}
\tag{36}
$$

Equation 36 means $f_0(\mathbf{x}^*) \leq f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \lambda_i^* f_i(\mathbf{x}^*) + \sum_{j=1}^{m} \nu_j^* h_j(\mathbf{x}^*) \leq f_0(\mathbf{x}^*)$ and hence implies

$$
f_0(\mathbf{x}^*) = g(\lambda^*, \nu^*) = f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \lambda_i^* f_i(\mathbf{x}^*) + \sum_{j=1}^{m} \nu_j^* h_j(\mathbf{x}^*)
\tag{37}
$$

As a result, the following condition must be satisfied:

$$
\begin{cases}
\sum_{i=1}^{n} \lambda_i^* f_i(\mathbf{x}^*) = 0 \\
\sum_{j=1}^{m} \nu_j^* h_j(\mathbf{x}^*) = 0
\end{cases}
\tag{38}
$$

Since for any feasible $\mathbf{x}$, $h_j(\mathbf{x}) = 0 \, (j = 1, 2, \cdots m)$, $\sum_{j=1}^{m} \nu_j^* h_j(\mathbf{x}^*) = 0$ always hold. Furthermore, since $\lambda_i^* f_i(\mathbf{x}^*) \leq 0$ for each i, $\sum_{i=1}^{n} \lambda_i^* f_i(\mathbf{x}^*) = 0$ if and only if for each i, there is $\lambda_i^* f_i(\mathbf{x}^*) = 0$.

Additionally, if $f_i(\mathbf{x}) \, (i = 0, 1, \cdots n)$ and $h_j(\mathbf{x}) \, (j = 1, 2, \cdots m)$ are differentiable. The Equation 37 basically tell us $\mathbf{x}^*$ is the minimizer of the Lagrangian $\mathcal{L}(\mathbf{x}, \lambda^*, \nu^*)$ and hence $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*, \nu^*) = 0$:

$$
\begin{aligned}
\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*, \nu^*) &= \nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \lambda_i^* \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{m} \nu_j^* \nabla h_j(\mathbf{x}^*) \\
&= 0
\end{aligned}
\tag{39}
$$

In conclusion, if strong duality hold for a optimization problem with optimal value $p^* = f_0(\mathbf{x}^*)$, there must be:

1. $\mathbf{x}^*$ must be feasible, hence, it must satisfied:

$$
\begin{cases}
f_i(\mathbf{x}^*) \leq 0 & i = 1, 2, \cdots n \\
h_j(\mathbf{x}^*) = 0 & j = 1, 2, \cdots m
\end{cases}
$$

2. $\lambda_i^* \geq 0$ for $i = 1, 2, \cdots n$

3. $\lambda_i^* f_i(\mathbf{x}^*)$ for $i = 1, 2, \cdots n$.

4. $\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \lambda_i^* \nabla f_i(\mathbf{x}^*) + \sum_{j=1}^{m} \nu_j^* \nabla h_j(\mathbf{x}^*) = 0$.

These conditions are called KKT conditions which give us the criteria to determine whether the strong duality hold. Particularly, if we can solve the equations listed in KKT conditions, we not only verify the strong duality hold, but find the optimal point $\mathbf{x}^*$ and λ^*, ν^* that make $g(\lambda^*, \nu^*) = p^*$.
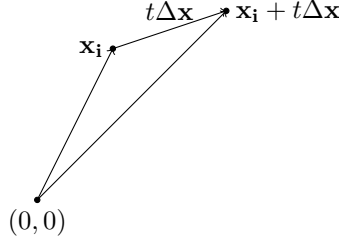
9   Optimization Algorithm

In general, we are looking for a iterative algorithm so that:

$$f(\mathbf{x_{i+1}}) \leq f(\mathbf{x_i})$$

In which, $\mathbf{x_{i+1}}$ and $\mathbf{x_i}$ satisfied:

$$\mathbf{x_{i+1}} = \mathbf{x_i} + t\Delta\mathbf{x} \tag{40}$$

with $t > 0$. The term $\Delta\mathbf{x}$ is a *direction vector* which indicate the incremental direction of $\mathbf{x_i}$. The term t indicate the incremental length along with the direction $\Delta\mathbf{x}$. A simple 2 dimensional illustration can be show as:

Now consider $f(\mathbf{x})$ is convex. According to the first-order condition, we have:

$$f(\mathbf{x_i} + \mathbf{t\Delta x}) \geq f(\mathbf{x_i}) + \nabla f(\mathbf{x_i})^T (t\Delta\mathbf{x_i})$$

We notice from this inequality that if $\nabla f(\mathbf{x_i})^T (t\Delta\mathbf{x_i}) \geq 0$, $f(\mathbf{x_i} + \mathbf{t\Delta x}) \geq f(\mathbf{x_i})$ always holds. Hence, we must choose t and $\Delta\mathbf{x}$ that satisfied:

$$\nabla f(\mathbf{x_i})^T t\Delta\mathbf{x} \leq 0$$

By considering $t > 0$, we notice the above inequality is equivalent to:

$$\nabla f(\mathbf{x_i})^T \Delta\mathbf{x} \leq 0 \tag{41}$$

Which means, we must choose the incremental direction that is opposite of the gradient direction.

9.1   Optimization under Approximation Perspective

For a convex function $f(\mathbf{x})$, we can use Taylor Approximation to approximate any point $\mathbf{y}$ base on a fixed point $\mathbf{x}$ as:

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T \nabla^2 f(\mathbf{z})(\mathbf{y} - \mathbf{x}) \tag{42}$$

Where $\mathbf{z}$ could be any point within $\mathbf{dom}f$. If we further consider f as a *strong convex* function, which means, there exist some α, so that the function $g(\mathbf{x})$:

$$g(\mathbf{x}) = f(\mathbf{x}) - \alpha\|\mathbf{x}\|_2^2 \text{ is convex}$$

Since $g(\mathbf{x})$ is convex, we have:

$$\nabla^2 g(\mathbf{x}) = \nabla^2 f(\mathbf{x}) - \alpha\mathbf{I} \succeq 0$$

Which implicit that for any $\mathbf{z} \in \mathbf{dom}f$, there is:

$$\nabla^2 f(\mathbf{z}) \succeq \alpha\mathbf{I}$$

We can apply this inequality to Equation 42 and hence:

$$
\begin{aligned}
f(\mathbf{y}) &= f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T \nabla^2 f(\mathbf{z})(\mathbf{y} - \mathbf{x}) \\
&\geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T \alpha\mathbf{I}(\mathbf{y} - \mathbf{x}) \\
&= f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2^2
\end{aligned}
\tag{43}
$$

Therefore, we successfully construct a lower bound of any point $\mathbf{y}$ in terms of a fix point $\mathbf{x}$. Now, we can substitute $\mathbf{y}$ by $\mathbf{x_{i+1}}$ and fix $\mathbf{x}$ as $\mathbf{x_i}$. In order to make $f(\mathbf{x_{i+1}}) \leq f(\mathbf{x_i})$, we need to choose a point $\mathbf{x_{i+1}}$ that can minimize $f(\mathbf{x_{i+1}})$. Indeed, we can choose the lower bound in Equation 43 as the approximation of

$f(\mathbf{x_{i+1}}) \approx f(\mathbf{x_i}) + \nabla f(\mathbf{x_i})^T(\mathbf{x_{i+1}} - \mathbf{x_i}) + \frac{\alpha}{2}\|\mathbf{x_{i+1}} - \mathbf{x_i}\|_2^2$ (Since we can think $\mathbf{x_i}$ as constant, the term $f(\mathbf{x_i})$ is also a constant) and then, minimize this approximation instead of minimizing the original $f(\mathbf{x_{i+1}})$ and construct a simple minimization problem:

$$\min_{\mathbf{x_{i+1}}} \quad f(\mathbf{x_i}) + \nabla f(\mathbf{x_i})^T(\mathbf{x_{i+1}} - \mathbf{x_i}) + \frac{\alpha}{2}\|\mathbf{x_{i+1}} - \mathbf{x_i}\|_2^2$$

The solution of this optimization is:

$$\mathbf{x_{i+1}} = \mathbf{x_i} - \frac{1}{\alpha}\nabla f(\mathbf{x_i})$$

Which is equivalent to Equation 40 by setting $t = \frac{1}{\alpha}$ and $\Delta\mathbf{x} = -\nabla f(\mathbf{x_i})$. Indeed, we can verify that by letting $\Delta\mathbf{x} = -\nabla f(\mathbf{x_i})$, Equation 41 always holds as:

$$\begin{aligned}
\nabla f(\mathbf{x_i})^T \Delta\mathbf{x} &= -\nabla f(\mathbf{x_i})^T \nabla f(\mathbf{x_i}) \\
&= -\|\nabla f(\mathbf{x_i})\|_2^2 \\
&\leq 0
\end{aligned}$$

In fact, the optimization method which choosing $-\nabla f(\mathbf{x_i})$ as incremental direction is called *Gradient Descend*.

9.2 Gradient Descend and Line Search

As we discussed before, by letting $\Delta\mathbf{x} = -\nabla f(\mathbf{x_i})$ in Equation 40, we get the Gradient Descend Method. After we have chosen the incremental direction, one more thing we need to concern is the value of t, or be more specific, the incremental length. We first notice that within the first-order condition:

$$f(\mathbf{x_i} + t\Delta\mathbf{x}) \geq f(\mathbf{x_i}) + t\nabla f(\mathbf{x_i})^T \Delta\mathbf{x_i}$$

If we force t to be large, the term $t\nabla f(\mathbf{x_i})^T\Delta\mathbf{x_i}$ could be very negative. As a result, the first-order condition can still hold even though $f(\mathbf{x_i}) \geq f(\mathbf{x_i} + t\Delta\mathbf{x})$, which against our intention: $f(\mathbf{x_i}) \leq f(\mathbf{x_i} + t\Delta\mathbf{x})$. On the other hand, if we choose t to be too small, the term $\mathbf{x_i} + t\Delta\mathbf{x}$ is almost identical to original $\mathbf{x_i}$, which means, the algorithm converge very slowly. The process that find the value of t is called *line search*. One general idea for addressing line search is that we can first assign t with an initial value, *e.g.* $t_{init} = 1$, then reduce the value of t iterative until it satisfied some constrain. This idea left us with 2 question:

1. what is the stop constrain

2. what is the decrease size of t in each iteration

For the first question, we first notice that we can approximate $f(\mathbf{x_i} + t\Delta\mathbf{x})$ by using first order Taylor Approximation:

$$f(\mathbf{x_i} + t\Delta\mathbf{x}) \approx f(\mathbf{x_i}) + t\nabla f(\mathbf{x_i})^T \Delta\mathbf{x}$$

The above equation hold when t is very small. Since $\Delta\mathbf{x} = -\nabla f(\mathbf{x_i})$, we can multiple the term $t\nabla f(\mathbf{x_i})^T\Delta\mathbf{x}$ with an $\alpha < 1$ and get:

$$f(\mathbf{x_i}) + t\nabla f(\mathbf{x_i})^T \Delta\mathbf{x} \leq f(\mathbf{x_i}) + t\alpha\nabla f(\mathbf{x_i})^T \Delta\mathbf{x}$$

Combine these two equations, we can conclude that for any $0 < \alpha < 1$, there exist a small enough t that satisfied:

$$f(\mathbf{x_i} + t\Delta\mathbf{x}) \leq f(\mathbf{x_i}) + t\alpha\nabla f(\mathbf{x_i})^T \Delta\mathbf{x} \tag{44}$$

Indeed, Equation 44 give us a good stopping constrain for line search, we can select an α and reduce the value of t iterative until this equation is satisfied. For the second question, we can reduce t by multiplying it with a β which satisfied $0 < \beta < 1$.

This line search algorithm is called *backtrack line search* and can be summarized as:

1. Choose the value of α and β

2. Choose the initial value of t

3. Multiple t with β and assign t with this new value($t := \beta t$) until Equation 44 is satisfied.

4. Return the value of t.

9.3 Newton's Method

According to the above discussion, we have noticed that we can minimize $f(\mathbf{x}+\Delta\mathbf{x})$ by minimize it's approximation. Particularly, by minimizing the first-order Taylor Approximation of $f(\mathbf{x}+\Delta\mathbf{x})$, we obtain Gradient Descent algorithm. Indeed, instead of using first-order approximation, we can use second-order Taylor Approximation of $f(\mathbf{x}+\Delta\mathbf{x})$. Thus, we have:

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

By minimizing this approximation over $\Delta\mathbf{x}$(in here, $\mathbf{x}$ is the initial point we choose hence, is a constant), we have:

$$\Delta\mathbf{x} = -\nabla^2 f(\mathbf{x})^{-1}\nabla f(\mathbf{x})$$

9.4 Steepest Descent

The Gradient Descent Algorithm we introduced before is a special case of a more general algorithm called *Steepest Descent*. As we discussed before, for a initial point $\mathbf{x}$, we want to find an incremental direction $\Delta\mathbf{x}$ and a incremental size t so that: $f(\mathbf{x} + t\Delta\mathbf{x}) \leq f(\mathbf{x})$. Be more specific, we want to minimize $f(\mathbf{x} + t\Delta\mathbf{x})$. In order to do so, we can minimize the approximation of $f(\mathbf{x} + t\Delta\mathbf{x})$ instead of the original function. Moreover, according to the first-order Taylor Approximation, we have:

$$f(\mathbf{x} + t\Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T t\Delta\mathbf{x}$$

Since $\mathbf{x}$ is an initial point and t is the incremental size that can be found through line search, we can treat these $\mathbf{x}$ and t as constant. Therefore, in order to minimize the approximation function, we only need to minimize the term $\nabla f(\mathbf{x})\Delta\mathbf{x}$ (as we have treated $f(\mathbf{x})$ and t as constants). As a result, we can find $\Delta\mathbf{x}$ by solving following equation:

$$\Delta\mathbf{x} = \underset{\Delta\mathbf{x}}{\arg\min}\, \nabla f(\mathbf{x})^T \Delta\mathbf{x}$$

We then notice that $\nabla f(\mathbf{x})^T \Delta\mathbf{x}$ is a affine(linear) function, which means, it can reach $-\infty$ if Δ is unconstrained. Additionally, the first-order Taylor Approximation only hold when $t\Delta\mathbf{x}$ is small enough, therefore, if $\Delta\mathbf{x}$ is unconstrained and being too large, we cannot use $f(\mathbf{x}) + \nabla f(\mathbf{x})^T t\Delta\mathbf{x}$ to approximate the original function and hence, the minimization of $f(\mathbf{x}) + \nabla f(\mathbf{x})^T t\Delta\mathbf{x}$ is useless under this situation. As a result, it is important to add some constrain over $\Delta\mathbf{x}$. The common constrain is restricting the norm of $\Delta\mathbf{x}$. Therefore, we can find $\Delta\mathbf{x}$ by solving the following optimization problem:

$$\min_{\Delta\mathbf{x}} \quad \nabla f(\mathbf{x})^T \Delta\mathbf{x}$$
$$s.t. \quad \|\Delta\mathbf{x}\| \leq 1$$

The $\|\Delta\mathbf{x}\|$ could mean any order norm, *e.g.* first order norm $\|\Delta\mathbf{x}\|_1$, second order norm $\|\Delta\mathbf{x}\|_2$.

10 Optimization Algorithm with Inequality Constrain

The inequality constrains within optimization problem are usually the most difficulty part to deal with. Therefore, finding a method to eliminate these constrain within optimization problems will help solve these problems. Indeed, for an optimization problem:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x})$$
$$s.t. \quad f_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \cdots, n$$
$$\quad\quad h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots, m$$

We can think the inequality constrain serve the following purpose: The inequality constrains define a set of $\mathbf{x}$, if $\mathbf{x}$ take value out of this set, the objective function($f_0(\mathbf{x})$) will take value ∞. Therefore, we can embed the inequality constrains into objective function by replacing them with the *indicator function* defined as:

$$\mathcal{I}(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0 \end{cases}$$

Being more specific, we can reconstruct the optimization problem with indicator function as:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x}) + \sum_{i=1}^{n} \mathcal{I}(f_i(\mathbf{x}))$$
$$s.t. \quad h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots, m$$

Following the definition of indicator function, it can be seen that the new objective function $f_0(\mathbf{x}) + \sum_{i=1}^{n} \mathcal{I}(f_i(\mathbf{x}))$ will goes to infinity if $f_i(\mathbf{x}) \leq 0$ is not satisfied and will equal to the original objective function $f_0(\mathbf{x})$ if the inequality constrains are hold.

Even though we successfully remove the inequality constrains, it is still difficulty to deal with the indicator function as it is non differentiable. A better idea is using some approximation function to replace indicator function. Particularly, a good approximation function $\phi(u)$ need to have the similar property as indicator function: $\phi(u)$ will approximate 0 if $u \leq 0$ and approximate ∞ if $u > 0$. One such function is *log-barrier* function defined as:

$$\phi(u) = -\frac{1}{t} \log(-u)$$

Where t is a constant chosen by us. By applying this approximation to the original problem, we have:

$$\min_{\mathbf{x}} \quad f_0(\mathbf{x}) + \frac{1}{t} \sum_{i=1}^{n} -\log(-f_i(\mathbf{x}))$$

$$s.t. \quad h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots, m$$

For our convenience, we can defined:

$$\Phi(\mathbf{x}) = \sum_{i=1}^{n} -\log(-f_i(\mathbf{x}))$$

Recalling the equivalent problem construction methodology, we can construct a equivalent problem by multiplying the original objective function with any constant since this action will not effect the optimal point. Therefore, we can construct an equivalent problem by multiplying the objective function in above optimization problem with constant t and get:

$$\min_{\mathbf{x}} \quad t f_0(\mathbf{x}) + \Phi(\mathbf{x})$$

$$s.t. \quad h_j(\mathbf{x}) = 0 \quad j = 1, 2, \cdots, m$$

As a result, we can solve the original optimization by solving this approximation problem. Now, let us have a closer look of log-barrier method by considering the stand form convex optimization problem. Following our discussion above, we can construct the approximation problem as:

$$\min_{\mathbf{x}} \quad t f_0(\mathbf{x}) + \Phi(\mathbf{x})$$

$$s.t. \quad \mathcal{A}\mathbf{x} = \mathbf{b}$$

The first step to solve this problem is constructing its Lagrangian:

$$\mathcal{L}(\mathbf{x}, \nu) = t f_0(\mathbf{x}) + \Phi(\mathbf{x}) + \nu^T(\mathcal{A}\mathbf{x} - \mathbf{b})$$

Now, suppose this problem has optimal point $\mathbf{x}^*$. This optimal point must be the minimizer of Lagrangian, which means, there must exist a ν^* so that:

$$\nabla \mathcal{L}(\mathbf{x}^*, \nu^*) = 0$$

By noticing that

$$\nabla \Phi(\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{-f_i(\mathbf{x})} \nabla f_i(\mathbf{x})$$

We have:

$$\nabla \mathcal{L}(\mathbf{x}^*, \nu^*) = t \nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \frac{1}{-f_i(\mathbf{x}^*)} \nabla f_i(\mathbf{x}^*) + \mathcal{A}^T \nu \tag{45}$$

$$= 0$$

This equation indicates that the approximation problem has optimal solution if and if this equation hold. Indeed, we can multiple Equation 45 with $\frac{1}{t}$ and get:

$$\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \frac{1}{-t f_i(\mathbf{x}^*)} \nabla f_i(\mathbf{x}^*) + \mathcal{A}^T \frac{\nu}{t} = 0$$

It is not hard to find out that the term $\nabla f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \frac{1}{-t f_i(\mathbf{x}^*)} \nabla f_i(\mathbf{x}^*) + \mathcal{A}^T \frac{\nu}{t}$ is the first-order derivative of the Lagrangian $\hat{\mathcal{L}}(\mathbf{x}, \hat{\lambda}, \hat{\nu})$ of the original optimization problem with $\hat{\lambda}_i = \frac{1}{-t f_i(\mathbf{x})}$ and $\hat{\nu} = \frac{\nu}{t}$. As a result, the

Equation 45 also indicates $\mathbf{x}^*$ minimize $\hat{\mathcal{L}}(\mathbf{x}, \hat{\lambda}, \hat{\nu})$ and hence, the Lagrange Dual Function of original optimization function can be obtained as:

$$g(\hat{\lambda}, \hat{\nu}) = f_0(\mathbf{x}^*) + \sum_{i=1}^{n} \hat{\lambda}_i f_i(\mathbf{x}^*) + \hat{\nu}^T (\mathcal{A}\mathbf{x}^* - \mathbf{b})$$

By substituting $\hat{\lambda}_i = \frac{1}{-t f_i(\mathbf{x}^*)}$ to this equation and notice $\mathcal{A}\mathbf{x}^* - \mathbf{b} = 0$, we have:

$$g(\hat{\lambda}, \hat{\nu}) = f_0(\mathbf{x}^*) - \frac{n}{t}$$

Where n is the number of inequality constrains. One important thing to note here is that the pair $(\hat{\lambda}, \hat{\nu})$ do not maximize the Lagrange Dual Function, therefore, we have:

$$p^* \geq g(\hat{\lambda}, \hat{\nu}) = f_0(\mathbf{x}^*) - \frac{n}{t}$$

Where p^* is the optimal value of the original problem. As a result, we have:

$$f_0(\mathbf{x}^*) - p^* \leq \frac{n}{t}$$

Which measure the margin between the optimal value of approximation problem $f(\mathbf{x}^*)$ and the optimal value of original problem p^*.

11 Application

A lot of real-world problems can be convert into convex optimization problems. In this part, we will introduce some examples of this.

11.1 Approximation Problem

In real world, we usually need to replace the value of some variables with some observed data. Being more specific, the value of y_i can be replaced by a set of observed data denoted as $\mathbf{a}_i = \begin{bmatrix} a_{i,1} & a_{i,2} & \cdots & a_{i,m} \end{bmatrix}^T$. We can replace the value of y_i with $\mathbf{a}_i$ by constructing an approximation function $f(\mathbf{a}_i)$ with parameter $\mathbf{x}$, $e.g.$ $y_i \approx f(\mathbf{a}_i) = \mathbf{a}_i^T \mathbf{x}$. Now, suppose we have several such y_i: $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T$ and observed data set:

$$\mathcal{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix}$$

we can approximate $\mathbf{y}$ as:

$$\mathbf{y} \approx \mathcal{A}\mathbf{x}$$

It is not hard to figure out that the error of the approximation can be measured as:

$$\epsilon = \mathbf{y} - \mathcal{A}\mathbf{x}$$

In which, each component of ϵ is $\epsilon_i = y_i - \mathbf{a}_i^T \mathbf{x}$. In order to construct a good approximation of $\mathbf{y}$, we want the magnitude of this error to be small, which means, we can estimate the parameters of this approximation by solving the following convex optimization problem:

$$\min_{\mathbf{x}} \quad \|\mathbf{y} - \mathcal{A}\mathbf{x}\|$$

In here, we measure the magnitude of error through its norm $\|\epsilon\|$. Indeed, we can denote the measurement in a more general form: $\Phi(\epsilon) = \sum_{i=1}^{n} \phi(\epsilon_i)$ and construct the optimization problem as:

$$\min_{\mathbf{x}, \epsilon} \quad \sum_{i=1}^{n} \phi(\epsilon_i)$$
$$s.t. \quad \epsilon = \mathbf{y} - \mathcal{A}\mathbf{x}$$

The function $\phi(\epsilon_i)$ here is called *penalty function* and we can treat it as the measurement of how the single error component ϵ_i contribute to the total error measurement $\Phi(\epsilon)$, *i.e.* the greater the value $\phi(\epsilon_i)$, the greater it contribute to the total error measurement $\Phi(\epsilon)$. As a result, the minimization will put more effort to minimize $\phi(\epsilon_i)$ and hence, reduce ϵ_i. In order to have a deeper understanding about this concept, we can think of two penalty function: $\phi(\epsilon_i) = \epsilon_i^2$ and $\phi_{\epsilon_i} = |\epsilon_i|$. If ϵ_i is large, the value of ϵ_i^2 is also large and we have $\epsilon_i^2 >> |\epsilon_i|$. Therefore, compare with $\sum_{i=1}^{n} |\epsilon_i|$, the minimization of $\sum_{i=1}^{n} \epsilon_i^2$ will put more effort to eliminate large ϵ_i. On the other hand, for small ϵ_i, we have $|\epsilon_i| << \epsilon_i^2$. As a result, the minimization of $\sum_{i=1}^{n} |\epsilon_i|$ will put weight on reducing small ϵ_i than the minimization $\sum_{i=1}^{n} \epsilon_i^2$.