

Optimization for Machine Learning

Pablo Mollá Chárlez

Contents

1	Introduction	3
2	Optimization Problem	3
2.1	Formal Definition	3
2.2	Challenges in Optimization	3
3	Sets and Functions	4
3.1	Convex Sets	4
3.2	Convex Functions	4
3.2.1	Definition 1	4
3.2.2	Definition 2	4
3.2.3	Strong Convexity	4
3.2.4	Properties of Convex Functions	4
3.3	Gradients	5
3.4	Hessians	5
3.4.1	Least Square Function: Convexity, Gradient and Hessian	5
3.5	Taylor's Expansion	6
3.5.1	Taylor's Expansion in \mathbb{R}	6
3.5.2	Taylor's Expansion in \mathbb{R}^n	6
3.6	Descent Direction	7
3.6.1	Descent Direction Lemma	7
3.6.2	Steepest Descent Direction	7
3.6.3	Gradient Descent Algorithm	7
3.6.4	L-Smoothness	7
3.6.5	Condition Number	8
3.6.6	Level Sets	8
4	Convergence Rates	8
4.1	Definition and Importance	8
4.2	Types of Convergence	8
4.2.1	Linear Convergence:	8
4.2.2	Quadratic Convergence:	8
4.3	Theorem: Convergence Rate of Gradient Descent for Strongly Convex Functions	9
4.4	Theorem: Convergence of Gradient Descent for Smooth and Convex Functions	9
4.5	Rate Convergence Algorithms	9
4.5.1	Gradient Descent (GD)	9
4.5.2	Classical Momentum	9
4.5.3	Nesterov Accelerated Gradient Descent (NAG)	10
4.5.4	Newton's Method	10
4.5.5	Gradient Descent with Step Size Chosen by Armijo's Rule	10

5	Continuous Optimization	11
5.1	Unconstrained vs. Constrained Optimization	11
5.1.1	Unconstrained Optimization:	11
5.1.2	Constrained Optimization:	11
6	Constrained Optimization	11
6.1	Definitions	11
6.2	Optimization with Equality Constraints	12
6.2.1	Theorem: First-Order Optimality Conditions	12
6.3	Optimization with Equality and Inequality Constraints	12
6.3.1	Theorem : KKT Conditions	12
6.3.2	Dual Problem	12
6.3.3	Strong & Weak Duality	12
6.3.3.1	Theorem: Slater's Condition	12
6.4	Example 1	13
6.5	Example 2	14

1 Introduction

Optimization is a fundamental aspect of **machine learning** and **data science**, playing a critical role in **model training** and **parameter tuning**. It involves finding the best solution for a given problem by **minimizing** or **maximizing** an objective function **while satisfying specific constraints**. From simple regression models to complex deep learning architectures, **optimization techniques ensure efficient learning and convergence**. This summary outlines the key components of optimization problems and techniques.

2 Optimization Problem

2.1 Formal Definition

- **Objective Function:** The function $f(x)$ that we aim to maximize or minimize. For example, in machine learning, this could be a prediction error.
- **Constraints:** Conditions that the solutions must satisfy, including:
 - **Inequality Constraints:** $g_i(x) \leq 0$
 - **Equality Constraints:** $h_j(x) = 0$

- **Feasible Set:** The set of all possible solutions that satisfy the constraints:

$$S = \{x \in \text{Dom } f \mid g_i(x) \leq 0 \ \forall i, \ h_j(x) = 0 \ \forall j\}.$$

- **Optimal Solution:** The solution x^* that maximizes or minimizes the objective function within the feasible set:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} h_j(x) = 0, & \forall j = 1, \dots, \ell, \\ g_i(x) \leq 0, & \forall i = 1, \dots, q. \end{cases}$$

The optimal solution is given by:

$$x^* = \arg \min_{x \in \mathbb{R}^n} f(x).$$

2.2 Challenges in Optimization

- **Global vs. Local Optima:** A **global optimum** represents the absolute best solution, whereas a **local optimum** is only the best within a specific neighborhood of the domain of f . Non-convex problems often cause algorithms to get trapped in local optima.
- **Uniqueness:** While a **unique solution** is ideal, **non-convex problems frequently have multiple optima**, necessitating careful consideration in selecting the most appropriate solution.
- **Multimodality:** Multimodal functions, characterized by **multiple peaks and valleys**, pose significant challenges. Algorithms such as **gradient descent may converge to a local rather than the global optimum**.
- **Convexity:** **Convex functions** simplify optimization, as **any local minimum is also a global minimum**. Non-convex functions, however, lack this property, making them harder to optimize.
- **Differentiability:** **Differentiable functions**, with smooth and continuous slopes, **are well-suited for gradient-based methods**. Non-differentiable points, such as sharp corners or sudden changes, require alternative approaches like subgradient methods.
- **Curse of Dimensionality:** As the **number of variables increases**, the **complexity of the problem grows significantly**, making it more challenging to visualize, analyze, and solve.
- **Non-separability:** **Dependencies between optimization variables prevent them from being optimized independently**, complicating the optimization process and often requiring joint optimization strategies.

- **Ill-posedness:** An ill-posed problem is one where the solution does not depend continuously on the data, meaning small changes in input can lead to large variations in the output, or the solution might not even exist or be unique. Ill-posedness can lead to instability in optimization, especially in numerical methods.
- **Ill-conditioning:** The function may have a poorly scaled gradient, meaning small changes in certain directions lead to disproportionately large changes in others. This imbalance can cause slow or unstable convergence during optimization, as it makes it difficult for gradient-based methods to find a stable path to the optimum.

3 Sets and Functions

3.1 Convex Sets

A set $S \subset \mathbb{R}^n$ is **convex** if, for any $x, y \in S$ and $\lambda \in [0, 1]$, $\lambda x + (1 - \lambda)y \in S$. This means that the line segment connecting any two points within the set lies entirely inside the set.

3.2 Convex Functions

3.2.1 Definition 1

A function f is **convex** if, for any x, y and $\lambda \in [0, 1]$, $f(\lambda \cdot x + (1 - \lambda) \cdot y) \leq \lambda \cdot f(x) + (1 - \lambda) \cdot f(y)$. This implies that the line segment connecting any two points on the graph of f lies above the graph itself. A **twice-differentiable function of a single variable is convex** if and only if its **second derivative is non-negative on its entire domain**.

3.2.2 Definition 2

A differentiable function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be **convex** if and only if $\forall x, \theta \in \mathbb{R}^d$, we have that:

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

3.2.3 Strong Convexity

A differentiable function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be **μ -strongly convex** if, for all $x, \theta \in \mathbb{R}^d$,

$$f(x) \geq f(\theta) + \nabla f(\theta)^T (x - \theta) + \frac{\mu}{2} \|x - \theta\|^2,$$

where $\mu > 0$ is the strong convexity parameter.

3.2.4 Properties of Convex Functions

The following operations preserve convexity:

- **Non-Negative Weighted Sum:** If f_1 and f_2 are convex functions, then $\alpha f_1 + \beta f_2$ is also convex for $\alpha, \beta \geq 0$.
- **Composition Rules:** If f is convex and increasing, and g is convex, then the composition $f(g(x))$ is convex.
- **Jensen's Inequality:** If f is a convex function and X is a random variable, then: $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

3.3 Gradients

The **gradient of a differentiable function** $f(x)$, denoted as $\nabla f(x)$, is a vector of partial derivatives that represents the rate of change of f with respect to each variable. Mathematically, for $f : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T.$$

For a differentiable function $f(x)$, the **gradient** $\nabla f(x)$ **points in the direction of the steepest ascent**. To **minimize the function**, one moves in the direction of $-\nabla f(x)$, which is the **direction of steepest descent**.

3.4 Hessians

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a twice-differentiable scalar-valued function. The **Hessian** of f is a **square matrix of second-order partial derivatives**, defined as:

$$H(f) = \nabla^2 f(w) = \begin{bmatrix} \frac{\partial^2 f}{\partial w_1^2} & \frac{\partial^2 f}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 f}{\partial w_1 \partial w_n} \\ \frac{\partial^2 f}{\partial w_2 \partial w_1} & \frac{\partial^2 f}{\partial w_2^2} & \cdots & \frac{\partial^2 f}{\partial w_2 \partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial w_n \partial w_1} & \frac{\partial^2 f}{\partial w_n \partial w_2} & \cdots & \frac{\partial^2 f}{\partial w_n^2} \end{bmatrix}.$$

In compact notation:

$$(H(f))_{ij} = \frac{\partial^2 f}{\partial w_i \partial w_j}, \quad \forall i, j \in \{1, 2, \dots, n\}.$$

The Hessian provides information about the curvature of f :

- If $H(f)$ is **positive semi-definite** at z , f is convex at $z \iff \forall z \in \mathbb{R}^n, z^T \cdot H_f \cdot z \geq 0$.
- If $H(f)$ is negative definite at z , f is locally concave at z .
- If $H(f)$ is indefinite, z may be a saddle point.

3.4.1 Least Square Function: Convexity, Gradient and Hessian

Let's prove that the least square function is convex, i.e:

$$f(w) = \frac{1}{2} \|y - Xw\|^2,$$

where $y \in \mathbb{R}^m$, $X \in \mathbb{R}^{m \times n}$, and $w \in \mathbb{R}^n$. To prove $f(w)$ is convex, we need to show:

$$f(\lambda w_1 + (1 - \lambda)w_2) \leq \lambda f(w_1) + (1 - \lambda)f(w_2), \quad \forall w_1, w_2 \in \mathbb{R}^n, \lambda \in [0, 1].$$

We have:

$$f(\lambda w_1 + (1 - \lambda)w_2) = \frac{1}{2} \|y - X(\lambda w_1 + (1 - \lambda)w_2)\|^2.$$

Using the linearity of X , this becomes:

$$f(\lambda w_1 + (1 - \lambda)w_2) = \frac{1}{2} \|\lambda(y - Xw_1) + (1 - \lambda)(y - Xw_2)\|^2.$$

Using the property of the squared norm, $\|u + v\|^2 = \|u\|^2 + \|v\|^2 + 2u^T v$, we get:

$$f(\lambda w_1 + (1 - \lambda)w_2) = \frac{1}{2} (\lambda^2 \|y - Xw_1\|^2 + (1 - \lambda)^2 \|y - Xw_2\|^2 + 2\lambda(1 - \lambda)(y - Xw_1)^T (y - Xw_2)).$$

Next, we calculate:

$$\lambda f(w_1) + (1 - \lambda)f(w_2) = \lambda \cdot \frac{1}{2} \|y - Xw_1\|^2 + (1 - \lambda) \cdot \frac{1}{2} \|y - Xw_2\|^2.$$

This simplifies to:

$$\lambda f(w_1) + (1 - \lambda)f(w_2) = \frac{1}{2} (\lambda \|y - Xw_1\|^2 + (1 - \lambda) \|y - Xw_2\|^2).$$

Now, compute the difference:

$$\text{Difference} := f(\lambda w_1 + (1 - \lambda)w_2) - (\lambda f(w_1) + (1 - \lambda)f(w_2)).$$

Substituting the expressions, we get:

$$\begin{aligned} \text{Difference} &= \frac{1}{2} (\lambda^2 \|y - Xw_1\|^2 + (1 - \lambda)^2 \|y - Xw_2\|^2 + 2\lambda(1 - \lambda)(y - Xw_1)^T(y - Xw_2)) \\ &\quad - \frac{1}{2} (\lambda \|y - Xw_1\|^2 + (1 - \lambda) \|y - Xw_2\|^2). \end{aligned}$$

After some algebra and factoring out $\lambda(1 - \lambda)$, the difference becomes:

$$\text{Difference} = -\frac{\lambda(1 - \lambda)}{2} \|Xw_1 - Xw_2\|^2.$$

Since $\|Xw_1 - Xw_2\|^2 \geq 0$ and $\lambda(1 - \lambda) \geq 0$ for $\lambda \in [0, 1]$, the difference is non-positive. Hence:

$$f(\lambda w_1 + (1 - \lambda)w_2) \leq \lambda f(w_1) + (1 - \lambda)f(w_2).$$

This proves that $f(w)$ is convex. The [gradient](#) calculation details of the function can be found on the exercise sheet, although here you can find the results for the gradient and the [hessian](#) of the least square function.

- **Gradient:** $\nabla f(x) = -X^T \cdot (y - Xw)$
- **Hessian:** $H_f(w) = X^T X$

3.5 Taylor's Expansion

3.5.1 Taylor's Expansion in \mathbb{R}

Let k be a natural number, $x_0 \in \mathbb{R}$, and f a function that is k -times continuously differentiable on an interval containing x_0 and x . Taylor's theorem states that there exists some $\xi \in (x_0, x)$ such that:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(k)}(\xi)}{k!}(x - x_0)^k.$$

Implication: Taylor's theorem allows us to approximate $f(x)$ around x_0 using increasingly accurate terms based on the derivatives of f at x_0 .

3.5.2 Taylor's Expansion in \mathbb{R}^n

For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that is continuously twice differentiable, the Taylor approximation around a point $x_0 \in \mathbb{R}^n$ is given by:

$$f(x) = f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0) + R_3(x),$$

where $R_3(x)$ is the remainder term:

$$R_3(x) = O(\|x - x_0\|^3), \quad \text{which vanishes as } x \rightarrow x_0.$$

3.6 Descent Direction

The concept of a descent direction identifies directions \mathbf{d} in which the function f decreases locally. Let \mathbf{x} be a point in the domain of f such that $\nabla f(\mathbf{x}) \neq 0$, meaning \mathbf{x} is not a critical point of f . A vector $\mathbf{d} \in \mathbb{R}^n \setminus \{0\}$ is a descent direction for f at \mathbf{x} if there exists $\bar{\alpha} > 0$ such that:

$$f(\mathbf{x} + \alpha \mathbf{d}) < f(\mathbf{x}), \quad \forall \alpha \in (0, \bar{\alpha}).$$

This means that f strictly decreases along the half-line $\{\mathbf{x} + \alpha \mathbf{d} : \alpha > 0\}$ for sufficiently small step sizes $\alpha > 0$.

3.6.1 Descent Direction Lemma

Let x be a non-critical point of f ($\nabla f(x) \neq 0$), and $d \in \mathbb{R}^n - \{0\}$. If $\nabla f(x)^T d < 0$, then d is a descent direction for f at x .

Proof

Since f is differentiable, by the first-order Taylor expansion theorem, we can approximate $f(x + \alpha d)$ for small $\alpha > 0$ as:

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^T d + o(\alpha),$$

where $o(\alpha)$ represents higher-order terms that vanish as $\alpha \rightarrow 0$.

If $\nabla f(x)^T d < 0$, then for small $\alpha > 0$, the term $\alpha \nabla f(x)^T d$ is negative, implying:

$$f(x + \alpha d) < f(x).$$

Therefore, d is a descent direction for f at x .

3.6.2 Steepest Descent Direction

The (unnormalized) direction $\mathbf{d} = -\nabla f(x)$ (anti-gradient) is called the steepest descent direction of f at x , as it yields the greatest decrease in f .

3.6.3 Gradient Descent Algorithm

To minimize a differentiable function f , the Gradient Descent algorithm operates with the following sequence of iterates:

- **Initialization:** Start with an initial point $x^{(0)}$.
- **Iteration:** For $k = 0, 1, 2, \dots$:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)} = x^{(k)} - \alpha^{(k)} \nabla f(x^{(k)}),$$

Where:

- $\mathbf{d}^{(k)} = -\nabla f(x^{(k)})$: The descent direction (negative gradient).
- $\alpha^{(k)}$: The step size (learning rate).

The iterations continue until a stopping criterion is reached, such as when $\|\nabla f(x^{(k)})\|$ is sufficiently small or the change in $f(x^{(k)})$ becomes negligible.

3.6.4 L-Smoothness

A differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be L -smooth if and only if,

$$f(x) \leq f(\theta) + \nabla f(\theta)^T (x - \theta) + \frac{L}{2} \|x - \theta\|^2 \quad \forall \theta, x \in \mathbb{R}^d$$

where $L > 0$ is the smoothness parameter.

3.6.5 Condition Number

The **condition number** κ measures how "well-conditioned" the optimization problem is. When a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is both **L -smooth** and **μ -strongly convex**, we define its condition number κ as:

$$\kappa = \frac{L}{\mu},$$

where L is the smoothness constant and μ is the strong convexity constant. When $L = \mu$, the function is **perfectly conditioned** ($\sim \kappa = 1$). Besides, a small condition number $\kappa \approx 1$ results in fast convergence and, a large condition number $\kappa \gg 1$ leads to slow convergence and oscillations (zigzag). In terms of Linear Algebra, the **condition number of a matrix** A is given by:

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}},$$

where λ_{\max} and λ_{\min} are **the largest and smallest eigenvalues of A** .

3.6.6 Level Sets

Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the **level set of f** corresponding to a scalar $c \in \mathbb{R}$ is the set of all points $x \in \mathbb{R}^n$ such that:

$$L_c = \{x \in \mathbb{R}^n \mid f(x) = c\}.$$

4 Convergence Rates

4.1 Definition and Importance

The convergence rate describes **how quickly an optimization algorithm approaches the optimal solution**. It is an important metric for comparing algorithms, as faster convergence leads to fewer iterations and faster solutions.

4.2 Types of Convergence

4.2.1 Linear Convergence:

Linear convergence occurs when the error decreases by a constant fraction with each iteration. Formally, this can be expressed as:

$$\|x^{(k+1)} - x^*\| \leq c \cdot \|x^{(k)} - x^*\|,$$

where c is a constant that represents the rate of error decrease, satisfying $0 < c < 1$. A smaller c indicates faster convergence.

4.2.2 Quadratic Convergence:

Quadratic convergence occurs when the error decreases by the square of the previous error at each iteration, leading to very rapid convergence near the solution. Formally, this is expressed as:

$$\|x^{(k+1)} - x^*\| \leq c \cdot \|x^{(k)} - x^*\|^2,$$

where c is a positive constant ($c > 0$). **Quadratic convergence is significantly faster than linear convergence,** especially near the optimal solution.

4.3 Theorem: Convergence Rate of Gradient Descent for Strongly Convex Functions

Assume f is L -smooth and μ -strongly convex. For gradient descent with a fixed step size $\alpha_k = \frac{1}{L}$, the iterates $(x_k)_{k \geq 0}$ satisfy:

$$f(x_k) - f(x^*) \leq e^{-\frac{k\mu}{L}} \cdot (f(x_0) - f(x^*)),$$

where:

- x^* is the minimizer of f ,
- $\frac{\mu}{L}$ determines the rate of convergence and depends on the condition number $\kappa = \frac{L}{\mu}$.

Gradient descent therefore achieves exponential (linear in log-scale) convergence rate for strongly convex functions.

4.4 Theorem: Convergence of Gradient Descent for Smooth and Convex Functions

For a convex and L -smooth function f , gradient descent with a step size $\alpha = \frac{1}{L}$ satisfies:

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

where x^* is the minimizer of f . If f is only assumed to be smooth and convex, gradient descent with a constant step size $\alpha = \frac{1}{L}$ still converges, but at a slower rate (sublinear rate).

4.5 Rate Convergence Algorithms

4.5.1 Gradient Descent (GD)

The basic gradient descent algorithm updates the current value of x by moving against the gradient of the function f with a fixed step size α .

```
Initialize  $x = x_0$  (starting point)
Set learning rate  $\alpha$ 
Repeat until convergence:
    Compute:  $\nabla f(x)$ 
    Update:  $x_{k+1} = x_k - \alpha \cdot \nabla f(x)$ 
If convergence criterion is met:
    Break
```

4.5.2 Classical Momentum

Classical momentum introduces a velocity term v that helps the algorithm gain speed along the direction of the gradient and avoid oscillations.

```
Initialize  $x = x_0, v = 0$  (starting point and velocity)
Set learning rate  $\alpha$ , momentum  $\mu$ 
Repeat until convergence:
    Compute Gradient:  $\nabla f(x)$ 
    Update velocity:  $v = \mu \cdot v + \alpha \cdot \nabla f(x)$ 
    Update:  $x = x - v$ 
If convergence criterion is met:
    Break
```

4.5.3 Nesterov Accelerated Gradient Descent (NAG)

Nesterov's method is similar to classical momentum, but it looks ahead in the direction of the momentum to compute the gradient, resulting in faster convergence.

Initialize $x = x_0, v = 0$ (starting point and velocity)
Set learning rate α , **momentum** μ
Repeat until convergence:
 Compute the lookahead position: $x_{\text{hat}} = x - \mu \cdot v$
 Compute gradient at the lookahead position: $\nabla f(x_{\text{hat}})$
 Update velocity: $v = \mu \cdot v + \alpha \cdot \nabla f(x_{\text{hat}})$
 Update: $x = x - v$
If convergence criterion is met:
 Break

4.5.4 Newton's Method

Newton's method updates the current point x using both the gradient $\nabla f(x)$ and the Hessian $H(x) = \nabla^2 f(x)$ (the second derivative), allowing for faster convergence especially when the function is not well-behaved.

Initialize $x = x_0$ (starting point)
Set learning rate α
Repeat until convergence:
 Compute Gradient: $\nabla f(x)$
 Compute Hessian: $H(x) = \nabla^2 f(x)$
 Update: $x = x - H(x)^{-1} \cdot \nabla f(x)$
 If convergence criterion is met:
 Break

4.5.5 Gradient Descent with Step Size Chosen by Armijo's Rule

Armijo's rule dynamically adjusts the step size α based on the current gradient $\nabla f(x)$ and the function value $f(x)$. The step size is chosen to satisfy a decrease in the function value proportional to the gradient direction, and $c \in (0, 1)$.

Initialize $x = x_0$ (starting point)
Set α (learning rate), β (factor for decreasing learning rate)
and c (sufficient decrease condition)
Repeat until convergence:
 Compute Gradient: $\nabla f(x)$
 While: $f(x - \alpha \cdot \nabla f(x)) > f(x) - c\alpha \cdot \|\nabla f(x)\|^2$ (**Armijo condition**)
 Update: $\alpha = \beta \cdot \alpha$ (if while not satisfied α needs to be reduced)
 Update: $x = x - \alpha \cdot \nabla f(x)$
If convergence criterion is met:
 Break

These are the pseudo-codes for the algorithms. Each optimization method enhances basic gradient descent by incorporating momentum, second-order derivatives, or dynamic step size adjustment to improve convergence behavior, especially for non-convex or ill-conditioned functions.

5 Continuous Optimization

5.1 Unconstrained vs. Constrained Optimization

5.1.1 Unconstrained Optimization:

In **unconstrained optimization**, the goal is to minimize a function $f(x)$ over its domain $D = \text{dom } f$, **without any explicit constraints on x** :

$$\min_{x \in D} f(x).$$

For these problems, the feasible set is simply D , the domain of f .

5.1.2 Constrained Optimization:

If **restrictions are imposed on x** (e.g., $g_i(x) \leq 0$ for certain constraint functions $g_i(x)$), the problem becomes a **constrained optimization problem**, where **solutions must satisfy these additional conditions**.

6 Constrained Optimization

Now, we want to optimize a problem under certain constraints:

- **Objective:** Minimize or Maximize a function $f(x)$ subject to constraints.
- **General Form:** We will denote this form as the **Primal Form** or **Primal Problem**.

$$\begin{aligned} & - \min_{x \in \mathbb{R}^n} f(x) \\ & - \text{subject to: } g_i(x) \leq 0, \quad i = 1, \dots, m \\ & \quad \quad \quad h_j(x) = 0, \quad j = 1, \dots, p \end{aligned}$$

6.1 Definitions

- **Feasible Set:** The feasible set (or feasible region) is the set of all points that satisfy the constraints of an optimization problem. Formally, for a problem with constraints $g_i(x) \leq 0$ and $h_j(x) = 0$, the feasible set S is:

$$S = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, \quad h_j(x) = 0, \quad \forall i, j\}$$

- **Feasible Solution:** A feasible solution is any point $x \in S$ that satisfies all problem constraints.
- **Optimal solution:** If it exists, is a feasible solution that minimizes (or maximizes) the objective function within the feasible set.
- **Lagrangian Function:** The **Lagrangian function** is defined as:

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{j=1}^p \lambda_j h_j(x)$$

where λ_j are the **Lagrange multipliers**.

- **Dual Problem:** Minimize with respect to x and λ_j s the **Lagrangian** $\mathcal{L}(x, \lambda)$:
 - $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda)$
 - subject to: $g_i(x) \leq 0, \quad i = 1, \dots, m$
 $h_j(x) = 0, \quad j = 1, \dots, p$

6.2 Optimization with Equality Constraints

6.2.1 Theorem: First-Order Optimality Conditions

Let x^* be a local minimum of $f(x)$ subject to equality constraints $h_j(x) = 0$ for $j = 1, \dots, p$. If x^* is a **regular point** (the gradients $\nabla h_1(x^*), \dots, \nabla h_p(x^*)$ are linearly independent), there exist **Lagrange multipliers** $\lambda_1, \lambda_2, \dots, \lambda_p$ such that:

$$\nabla f(x^*) + \sum_{j=1}^p \lambda_j \nabla h_j(x^*) = 0, \quad h_j(x^*) = 0, \quad j = 1, \dots, p$$

6.3 Optimization with Equality and Inequality Constraints

The **Karush-Kuhn-Tucker (KKT)** conditions are necessary conditions to check optimality in **problems involving both equality and inequality constraints**. They extend the method of **Lagrange multipliers** to handle inequality constraints. The primal problem which aims at minimizing the objective function $f(x)$ is turned into the minimization of the following **Lagrangian function**:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i \cdot g_i(x) + \sum_{j=1}^p \mu_j \cdot h_j(x)$$

We call the **primal feasibility** the conditions that guarantee that the solution lies within the feasible region in the primal problem. Besides, we call **dual feasibility** the condition that forces the **Lagrange multipliers** associated with the inequality constraints must be non-negative.

6.3.1 Theorem : KKT Conditions

Let $f(x)$, $g_i(x)$, and $h_j(x)$ be continuously differentiable. If x^* is a **local minimum** and **satisfies certain regularity conditions**, then there exist $\lambda_i \geq 0$ and μ_j such that the **KKT** conditions hold.

6.3.2 Dual Problem

The **dual problem** is derived by minimizing the **Lagrangian** over x :

$$g(\lambda, \mu) = \inf_x \mathcal{L}(x, \lambda, \mu)$$

Then, the **dual problem** is:

$$\max_{\lambda \geq 0, \mu} g(\lambda, \mu)$$

6.3.3 Strong & Weak Duality

After solving both, the **primal problem** (x^*) and **dual problem** (λ, μ), we have the following 2 properties:

1. **Weak Duality:** $f(x^*) \geq g(\lambda, \mu)$
2. **Strong Duality:** $f(x^*) = g(\lambda^*, \mu^*)$

6.3.3.1 Theorem: Slater's Condition

For a **convex optimization problem** (objective function and inequality constraint functions are convex while the equality constraint functions are affine), if there exists a strictly feasible point x (one that satisfies $g_i(x) < 0$, $h_j(x) = 0$), then **strong duality holds**.

6.4 Example 1

Let's consider the following optimization problem we want to solve:

$$\text{Minimize: } f(x) = x^2 \quad \text{subject to: } g(x) = x - 2 \leq 0.$$

To handle the constraint, we define the **Lagrangian function**:

$$\mathcal{L}(x, \lambda) = f(x) + \lambda \cdot g(x),$$

Where:

- $f(x)$ is the objective function (x^2),
- $g(x)$ is the constraint ($x - 2$),
- $\lambda \geq 0$ is the **Lagrange multiplier**.

Then, in our case:

$$\mathcal{L}(x, \lambda) = x^2 + \lambda(x - 2).$$

The **dual function** $g(\lambda)$ is obtained by minimizing the **Lagrangian** over x :

$$g(\lambda) = \inf_x \mathcal{L}(x, \lambda).$$

Therefore, by minimizing $\mathcal{L}(x, \lambda)$ with respect to x :

$$\frac{\partial \mathcal{L}}{\partial x} = 2x + \lambda \longleftrightarrow 2x + \lambda = 0 \longleftrightarrow x = -\frac{\lambda}{2}$$

By substituting $x = -\frac{\lambda}{2}$ into $\mathcal{L}(x, \lambda)$ to compute $g(\lambda)$:

$$\mathcal{L}\left(-\frac{\lambda}{2}, \lambda\right) = \left(-\frac{\lambda}{2}\right)^2 + \lambda\left(-\frac{\lambda}{2} - 2\right) = \frac{\lambda^2}{4} - \frac{\lambda^2}{2} - 2\lambda = -\frac{\lambda^2}{4} - 2\lambda$$

The **dual function** has to be maximized $g(\lambda)$ subject to $\lambda \geq 0$:

$$\max_{\lambda \geq 0} g(\lambda) = -\frac{\lambda^2}{4} - 2\lambda.$$

Thus, we need to differentiate $g(\lambda)$ with respect to λ :

$$\frac{dg}{d\lambda} = -\frac{\lambda}{2} - 2 = 0 \longleftrightarrow \lambda = -4 \quad (\text{but this violates } \lambda \geq 0)$$

Therefore, $\lambda = 0$, and:

$$g(0) = -\frac{(0)^2}{4} - 2(0) = 0.$$

The **primal solution** gives $x^* = 2$ and $f(x^*) = 4$, the **dual solution** gives $g(\lambda) = 0$, which satisfies weak duality ($4 = f(x^*) \geq g(\lambda) = 0$). If Slater's Condition holds (i.e., $g(x) < 0$ for some x) (which doesn't because $g(2) = 0$), then strong duality ensures $f(x^*) = g(\lambda^*)$.

6.5 Example 2

We want to solve the following optimization problem:

$$\text{Minimize: } f(x_1, x_2) = x_1^2 + x_2^2$$

Subject to:

- Inequality constraint: $g(x_1, x_2) = x_1 + x_2 - 1 \leq 0$,
- Equality constraint: $h(x_1, x_2) = x_1 - x_2 = 0$.

The **Lagrangian** for this problem is defined as:

$$\mathcal{L}(x_1, x_2, \lambda, \mu) = f(x_1, x_2) + \lambda g(x_1, x_2) + \mu h(x_1, x_2) = x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1) + \mu(x_1 - x_2)$$

Where:

- $\lambda \geq 0$ is the **Lagrange** multiplier for the inequality constraint,
- μ is the **Lagrange** multiplier for the equality constraint.

The dual function $g(\lambda, \mu)$ is obtained by minimizing the **Lagrangian** over x_1 and x_2 :

$$g(\lambda, \mu) = \inf_{x_1, x_2} \mathcal{L}(x_1, x_2, \lambda, \mu).$$

Therefore, we need to compute the gradient of the **Lagrangian** with respect to x_1 and x_2 by taking partial derivatives of \mathcal{L} with respect to x_1 and x_2 :

- $\frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 + \lambda + \mu \iff \frac{\partial \mathcal{L}}{\partial x_1} = 0 \iff 2x_1 + \lambda + \mu = 0 \iff x_1 = -\frac{\lambda + \mu}{2}$
- $\frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 + \lambda - \mu \iff \frac{\partial \mathcal{L}}{\partial x_2} = 0 \iff 2x_2 + \lambda - \mu = 0 \implies x_2 = -\frac{\lambda - \mu}{2}$

By substituting $x_1 = -\frac{\lambda + \mu}{2}$ and $x_2 = -\frac{\lambda - \mu}{2}$ into \mathcal{L} , we would obtain the **dual function** that would need to be maximized:

$$\mathcal{L}\left(-\frac{\lambda + \mu}{2}, -\frac{\lambda - \mu}{2}, \lambda, \mu\right) = \left(-\frac{\lambda + \mu}{2}\right)^2 + \left(-\frac{\lambda - \mu}{2}\right)^2 + \lambda\left(-\frac{\lambda + \mu}{2} - \frac{\lambda - \mu}{2} - 1\right) + \mu\left(-\frac{\lambda + \mu}{2} + \frac{\lambda - \mu}{2}\right).$$