Preliminaries on Optimization

Adaptive and Cooperative Algorithms (ECE 457A)

ECE, MME, and MSCI Departments,
University of Waterloo, ON, Canada

Course Instructor: Benyamin Ghojogh

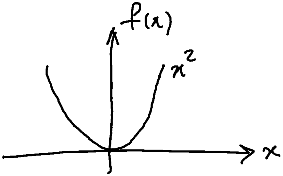
Fall 2023

What is Optimization?

Optimization problem

- Consider a function representing some cost. We call it **cost function** or **objective function**.
- We want to **minimize** or **maximize** this objective function.
- Examples:
  - Example for **minimization**: the cost function can be the error of some airplane structure from the perfect aerodynamic structure.
  - Example for **maximization**: the objective function can be the profit of the company.
  - **All life** is optimization!
  - **All machine learning** in artificial intelligence is optimization!
- The variables of the objective function are called the **objective variables** or **decision variables** or **optimization variables**.
- Example:

$$\underset{\boxed{x}}{\text{minimize}} \quad f(x) = x^2.$$



Univariate and multivariate optimization problems

- The optimization problem can be **univariate**, meaning that the optimization problem has only one scalar variable. Example:

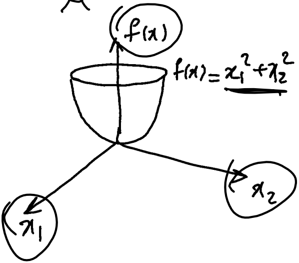
$$\underset{x}{\text{minimize}} \quad f(x) = x^2.$$

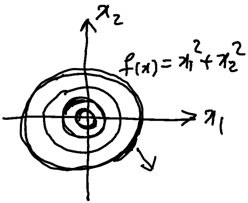
- The optimization problem can be **multivariate**, meaning that the optimization problem has several scalar variables $\{x_1, \dots, x_n\}$. These variables can be combined into a vector $\underbrace{\boldsymbol{x} = [x_1, \dots, x_n]^\top}$ or matrix. Example:

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{x} = \underbrace{x_1^2 + \cdots + x_n^2}.$$

A





Unconstrained and constrained problems

- The optimization problem can be **unconstrained**, meaning that we simply optimize a function only. Example:

$$\bigstar \quad \underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{x}.$$

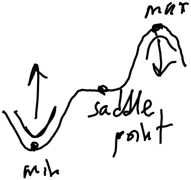
- The optimization problem can be **constrained**, meaning that we optimize a function while there are some constraints on the optimization variables. Example:

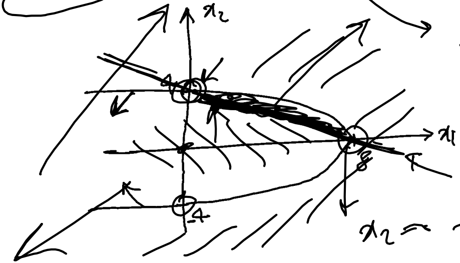
$$x^2 \rightarrow 2x \overset{set}{=} 0$$

$$\boxed{x \le 0}$$

$$\frac{\tan x}{\sqrt{\sin x^2}} \rightarrow$$

$$\begin{aligned} \underset{\boldsymbol{x}=[x_1,x_2]^\top}{\text{minimize}} \quad & \boxed{f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{x} = x_1^2 + x_2^2} \\ \text{subject to} \quad & x_1 + 2x_2 = 8, \\ \text{set.} \quad & 2x_1 + x_2^2 \le 16. \end{aligned}$$





$$x_1 = 0 \rightarrow x_2^2 \le 16$$

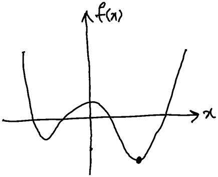
$$\downarrow$$

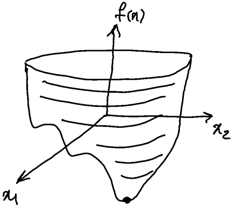
$$-4 \le x_2 \le 4$$

$$x_2 = 0 \rightarrow 2x_1 \le 16 \rightarrow x_1 \le 8$$

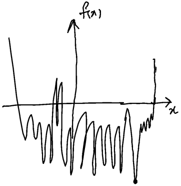
Optimization versus search

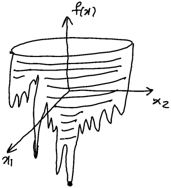
- If the objective problem is **simple enough**, we can solve it using **classic optimization** methods. We will learn important classic methods.

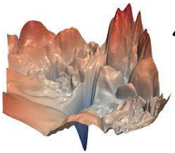




- If the objective function is **complicated** or if we have **too many constraints**, we can use **search** for finding a good solution.

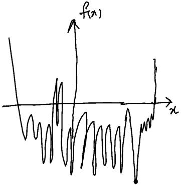


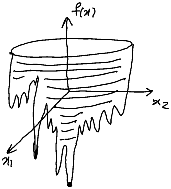


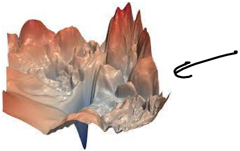


When to use search-based (metaheuristic) optimization

- When we have a **complicated** (highly con-convex) optimization landscape:



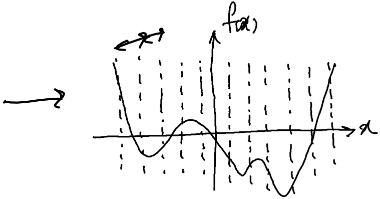


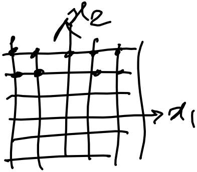


- or when the **gradient** of function is **hard to compute**,
- or when the function is not known but it works as a **black-box**, i.e., it outputs a value for each input fed to it.
- In these cases, we need:
  - either **non-convex optimization**,
  - or **search-based** optimization (**metaheuristic** optimization).

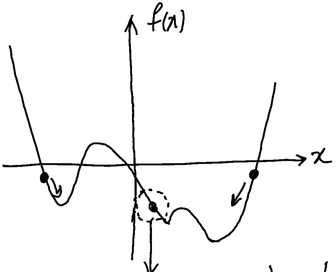
Search for optimization

- We can do **grid search** or **brute-force search**.

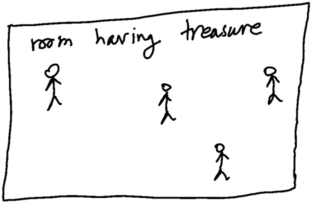




- Or we can **search wisely** by **metaheuristic optimization**. We will learn several important metaheuristic optimization methods.



better local ⇒ move toward it



Preliminaries on Metaheuristic Optimization

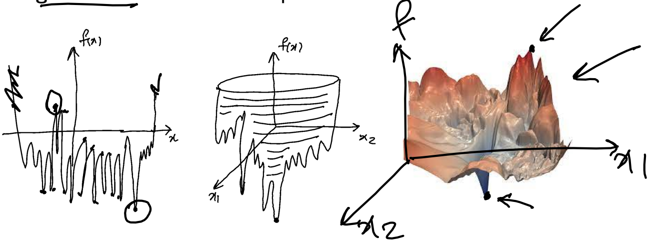
Heuristic and Metaheuristic Methods

- When the problem is complicated, a **heuristic** method approximates its solution.
- It gives a **good enough guess** of the solution to the problem, but that you may not really know **how good** it is.
- **Heuristics** are often **problem-dependent**, i.e., you define a heuristic for a given problem.
- **Metaheuristics** are **problem-independent** techniques that can be applied to a broad range of problems.
- **Metaheuristic optimization** methods can solve various complicated optimization problems using wise search.
- **Metaheuristic** methods are considered as a family of methods in **soft computing**.

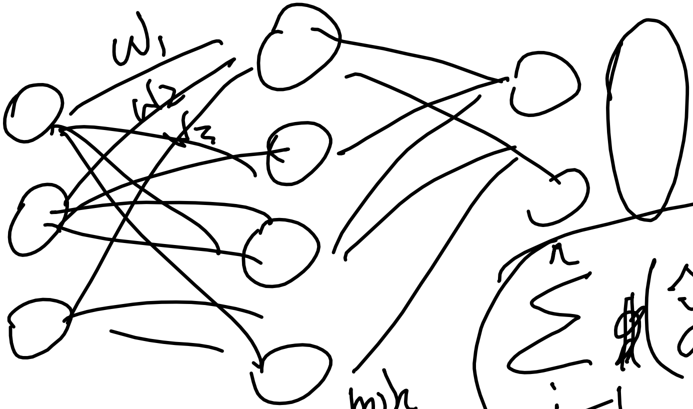
Exploration vs. Exploitation

Some things need to be defined:

- **Fitness** vs. **cost**: We usually minimize the cost function but maximize the fitness function. Fitness and cost can be converted to each other by changing maximization to minimization or vice versa.
- **Optimization landscape**: the optimization cost/fitness function
- **Local best** vs. **global best** in the landscape



- **Exploitation**: local search around the solution because the global optimum might be close to the current solution.
- **Exploration**: search far away from the solution (explore the landscape) because the global optimum might be far away from the current solution. It helps not to get **stuck in local optimum**.

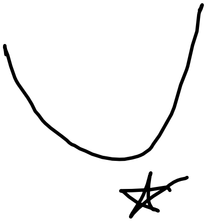


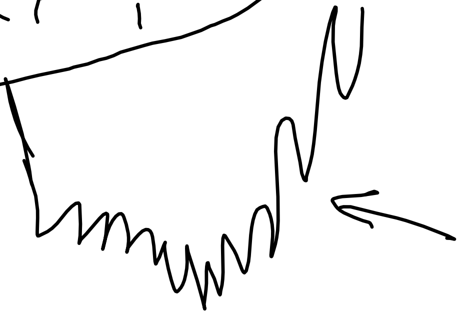
min

cost loss

$$\sum_{i=1}^{n} \left(\hat{y}_i - y_i \right)^2$$

min
$\{w_1, \dots w_{1000}\}$





Preliminaries on Sets and Norms

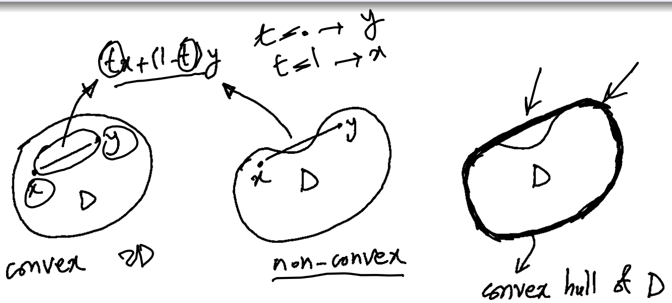
Convex set

Definition (Convex set and convex hull)

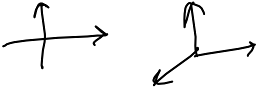
A set $\mathcal{D}$ is a convex set if it completely contains the line segment between any two points in the set $\mathcal{D}$:

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}, \underline{0 \leq t \leq 1} \implies \underbrace{t\boldsymbol{x} + (1-t)\boldsymbol{y}} \boxed{\in \mathcal{D}}.$$

The convex hull of a (not necessarily convex) set $\mathcal{D}$ is the smallest convex set containing the set $\mathcal{D}$. If a set is convex, it is equal to its convex hull.



Inner product



Definition (Inner product of vectors)

Consider two vectors $\underline{\boldsymbol{x} = [x_1, \dots, x_d]}^\top \boxed{\in \mathbb{R}^d}$ and $\underline{\boldsymbol{y} = [y_1, \dots, y_d]^\top} \boxed{\in \mathbb{R}^d}$. Their **inner product**, also called **dot product**, is:

$$\underline{\langle \boldsymbol{x}, \boldsymbol{y} \rangle} = \underline{\overset{\downarrow}{\boldsymbol{x}}^\top \boldsymbol{y}} = \underline{\sum_{i=1}^{d} x_i \, y_i}.$$

Definition (Inner product of matrices)

We also have inner product between matrices $\boxed{\boldsymbol{X}}, \boxed{\boldsymbol{Y}} \in \boxed{\mathbb{R}^{d_1 \times d_2}}$. Let $\boldsymbol{X}_{ij}$ denote the (i, j)-th element of matrix $\boldsymbol{X}$. The inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is:

$$\underline{\langle \boldsymbol{X}, \boldsymbol{Y} \rangle} = \overset{\downarrow}{\boxed{\mathbf{tr}}}(\underline{\boldsymbol{X}^\top \boldsymbol{Y}}) = \underline{\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}_{i,j} \, \boldsymbol{Y}_{i,j}},$$

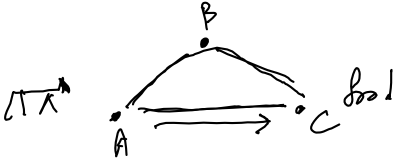
where $\mathbf{tr}(.)$ denotes the trace of matrix.

Norm

Definition (Norm)

A function $\|\cdot\| : \mathbb{R}^d \to \mathbb{R}$, $\|\cdot\| : \boldsymbol{x} \mapsto \|\boldsymbol{x}\|$ is a **norm** if it satisfies:

1. $\|\boldsymbol{x}\| \geq 0, \forall \boldsymbol{x}$
2. $\|a\boldsymbol{x}\| = |a|\|\boldsymbol{x}\|, \forall \boldsymbol{x}$ and all scalars a
3. $\|\boldsymbol{x}\| = 0$ if and only if $\boldsymbol{x} = 0$
4. Triangle inequality: $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$.



Important norms for vectors

Some important norms for a vector $\boldsymbol{x} = [x_1, \dots, x_d]^\top$ are as follows.

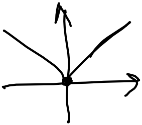
- The ℓ_p **norm** is:

$$\boxed{\|\boldsymbol{x}\|_p := \big(|x_1|^p + \dots + |x_d|^p\big)^{1/p},}$$

  where $p \geq 1$ and $|.|$ denotes the absolute value.

- Two well-known ℓ_p norms are ℓ_1 **norm** and ℓ_2 **norm** (also called the **Euclidean norm**) with $p = 1$ and $p = 2$, respectively:

$$\|\boldsymbol{x}\|_1 := |x_1| + \dots + |x_d| = \sum_{i=1}^{d} |x_i|,$$



$$\|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \dots + x_d^2} = \sqrt{\sum_{i=1}^{d} x_i^2},$$

- The ℓ_∞ **norm**, also called the **infinity norm**, the **maximum norm**, or the **Chebyshev norm**, is:

$$\boxed{\|\boldsymbol{x}\|_\infty := \max\{|x_1|, \dots, |x_d|\}.}$$

Important norms for matrices

Some important norms for a matrix $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$ are as follows.

- The formulation of the **Frobenius norm** for a matrix is similar to the formulation of ℓ_2 norm for a vector:

$$\boxed{\|\boldsymbol{X}\|_F := \sqrt{\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}_{i,j}^2},}$$

where $\boldsymbol{X}_{ij}$ denotes the (i, j)-th element of $\boldsymbol{X}$.

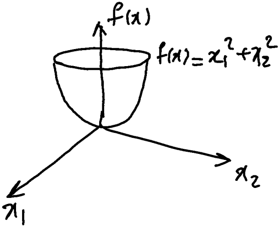
Quadratic forms using norms

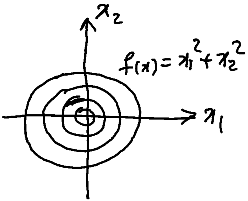
For $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$, we have:

$$\star \quad \underline{\|\boldsymbol{x}\|_2^2} = \underline{\boldsymbol{x}^\top \boldsymbol{x}} = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \sum_{i=1}^{d} x_i^2,$$

$$\star \quad \|\boldsymbol{X}\|_F^2 = \underline{\mathbf{tr}(\boldsymbol{X}^\top \boldsymbol{X})} = \langle \boldsymbol{X}, \boldsymbol{X} \rangle = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \boldsymbol{X}_{i,j}^2,$$

which are convex and in quadratic forms.





Preliminaries on Functions

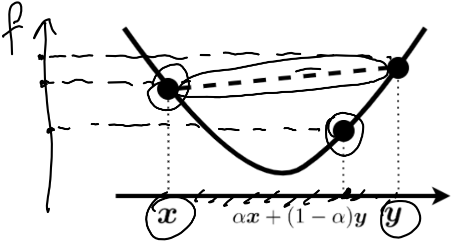
Convex function

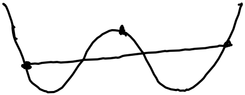
Definition (Convex function)

A function $f(.)$ with domain $\mathcal{D}$ is convex if:

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \boxed{\leq} \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}, \tag{1}$$

where $\alpha \in [0,1]$.





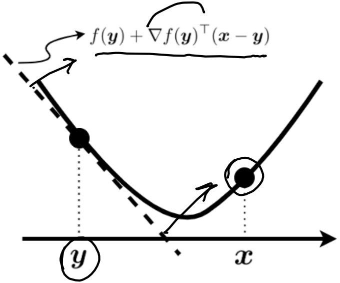
If $\overset{\leq}{\cancel{\leq}}$ is changed to $\overset{\geq}{\cancel{\geq}}$ in Eq. (1), the function is *concave*.

Convex function

Definition (Convex function)

If the function $f(.)$ is differentiable, it is convex if:

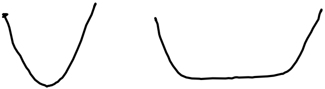
$$f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}. \tag{2}$$



$f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y})$
$\boldsymbol{y}$
$\boldsymbol{x}$

If $\geq$ is changed to $\leq$ in Eq. (2), the function is *concave*.

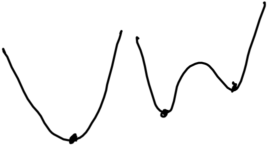
Convex function

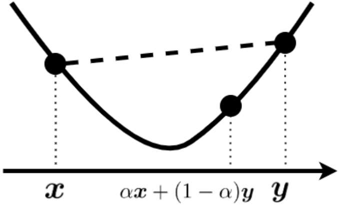


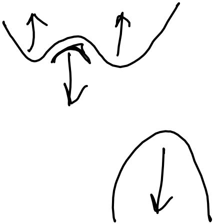
Definition (Convex function)

If the function $f(.)$ is twice differentiable, it is convex if its second-order derivative is positive semi-definite:

$$\overbrace{\nabla^2 f(\boldsymbol{x})}^{\text{matrix}} \boxed{\succeq} \mathbf{0}, \quad \forall \boldsymbol{x} \in \mathcal{D}. \tag{3}$$







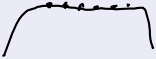
If $\succeq$ is changed to $\preceq$ in Eq. (3), the function is *concave*.

Preliminaries on Optimization

Min, max, sup, inf

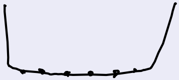
Definition (Minimum, maximum, infimum, and supremum)

A **minimum** and **maximum** of a function $f : \mathbb{R}^d \to \mathbb{R}$ $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, with domain $\mathcal{D}$, are defined as:



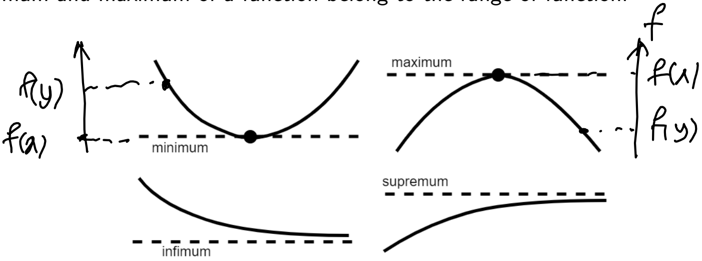
$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \le f(\boldsymbol{y}), \quad \forall \boldsymbol{y} \in \mathcal{D},$$

$$\max_{\boldsymbol{x}} f(\boldsymbol{x}) \ge f(\boldsymbol{y}), \quad \forall \boldsymbol{y} \in \mathcal{D},$$



respectively.

The minimum and maximum of a function belong to the range of function.



Min, max, sup, inf

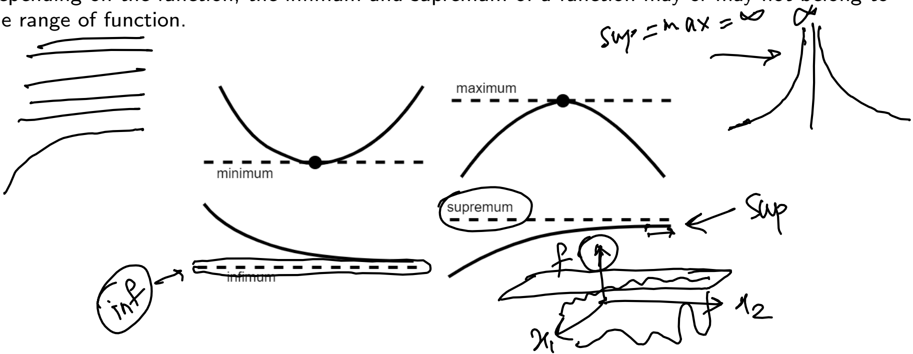
Definition (Infimum and supremum)

Infimum and **supremum** are the lower-bound and upper-bound of function, respectively:

$$\inf_{\boldsymbol{x}} f(\boldsymbol{x}) := \boxed{\max}\{z \in \mathbb{R} \mid \overbrace{\underbrace{z \le f(\boldsymbol{x})}}^{\text{lower bound}}, \forall \boldsymbol{x} \in \mathcal{D}\},$$

$$\sup_{\boldsymbol{x}} f(\boldsymbol{x}) := \boxed{\min}\{z \in \mathbb{R} \mid \underbrace{z \ge f(\boldsymbol{x})}_{\text{upper bound}}, \forall \boldsymbol{x} \in \mathcal{D}\}.$$

Depending on the function, the infimum and supremum of a function may or may not belong to the range of function.



Local and global minimizers

$\exists$: there exists

$\forall$: for all

Definition (Local minimizer)

A point $\boldsymbol{x} \in \mathcal{D}$ is a **local minimizer** of function $f(.)$ if and only if:

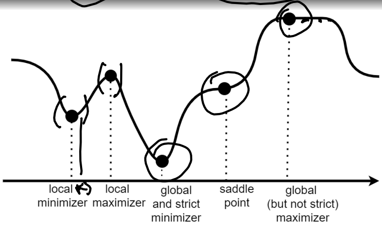
$$\underline{\boxed{\exists \epsilon} > 0 :} \; \underline{\forall \boldsymbol{y} \in \mathcal{D},} \; \underline{\|\boldsymbol{y} - \boldsymbol{x}\|_2} \boxed{\leq \epsilon} \implies \underline{f(\boldsymbol{x})} \leq \underline{f(\boldsymbol{y})}, \tag{4}$$

meaning that in an ϵ-neighborhood of $\boldsymbol{x}$, the value of function is minimum at $\boldsymbol{x}$.

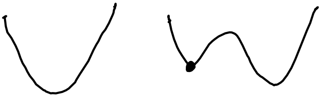
Definition (Global minimizer)

A point $\boldsymbol{x} \in \mathcal{D}$ is a **global minimizer** of function $f(.)$ if and only if:

$$\underline{f(\boldsymbol{x})} \leq f(\boldsymbol{y}), \; \boxed{\forall \boldsymbol{y} \in \mathcal{D}.} \tag{5}$$


local minimizer
local maximizer
global and strict minimizer
saddle point
global (but not strict) maximizer

Minimizer in convex function



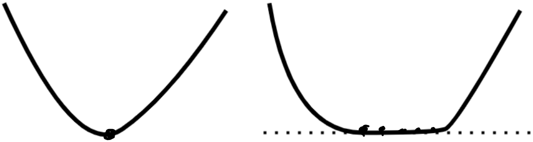
Lemma (Minimizer in convex function)

In a **convex function***, any local minimizer is a global minimizer. In other words, in a convex function, there exists only* **one** *local minimum value which is the global minimum value.*

Proof.

Proof can be found in the appendix of the tutorial [1]. □

As an imagination, a convex function is like a multi-dimensional bowl with only one minimum value (it may have several local minimizers but with the same minimum values).



strongly convex

convex (but not strongly convex)

Minimizer in convex function

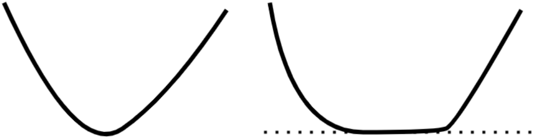
Lemma (Gradient of a convex function at the minimizer point)

When the function $f(.)$ is convex and differentiable, a point $\boldsymbol{x}^*$ is a minimizer if and only if:

$$\boxed{\nabla f(\boldsymbol{x}^*) = \mathbf{0}.} \longrightarrow \text{first order condition}$$

Proof.

Proof can be found in the appendix of the tutorial [1]. □



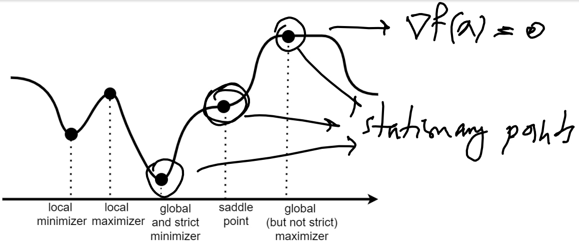
strongly convex

convex (but not strongly convex)

Stationary, extremum, and saddle points

Definition (Stationary, extremum, and saddle points)

- In a general (not-necessarily-convex) function $f(.)$, a point $\boldsymbol{x}^*$ is a **stationary** if and only if $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$.
- By passing through a **saddle point**, the sign of the second derivative flips to the opposite sign.
- **Minimizer** and **maximizer** points (locally or globally) minimize and maximize the function, respectively.
- A **saddle point** is neither minimizer nor maximizer, although the gradient at a saddle point is zero.
- Both minimizer and maximizer are also called the **extremum points**.
- A stationary point can be either a minimizer, a maximizer, or a saddle point of function.



local minimizer
local maximizer
global and strict minimizer
saddle point
global (but not strict) maximizer

$\nabla f(a) = 0$

stationary points

First-order optimality condition

Lemma (First-order optimality condition [2, Theorem 1.2.1])

If $\boldsymbol{x}^*$ is a local minimizer for a differentiable function $f(.)$, then:

$$\boxed{\nabla f(\boldsymbol{x}^*) = \mathbf{0}.} \tag{6}$$

Note that if $f(.)$ is convex, this equation is a necessary and sufficient condition for a minimizer.

Proof.

Proof can be found in the appendix of the tutorial [1]. □

Note

If setting the derivative to zero, $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$, gives a closed-form solution for $\boldsymbol{x}^*$, the optimization is done. Otherwise, we should solve it iteratively by either classic optimization or metaheuristic optimization.

Arguments of optimization

Definition (Arguments of minimization and maximization)

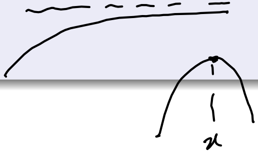
In the domain of function, the point which minimizes (resp. maximizes) the function $f(.)$ is the argument for the minimization (resp. maximization) of function.

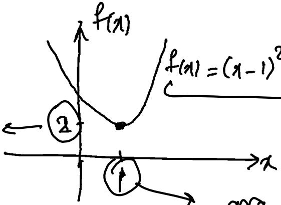
The minimizer and maximizer of function are denoted by

$$\arg\min_{\boldsymbol{x}} f(\boldsymbol{x}), \text{ and}$$

$$\arg\max_{\boldsymbol{x}} f(\boldsymbol{x}),$$

respectively.





$\underbrace{f(x) = (x-1)^2 + 2}_{} \Rightarrow \arg\min_x f(x) = 1$

$\min_x f(x) = 2$

$\min_x f(x) = 2$

$\arg\min_x f(x) = 1$

Converting optimization problems

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)}$$

Converting max to min and vice versa

We can convert convert maximization to minimization and vice versa:

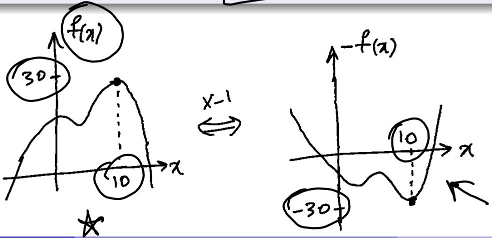
$$\boxed{\underset{\boldsymbol{x}}{\text{maximize}}\ f(\boldsymbol{x})} = \underbrace{-\underset{\boldsymbol{x}}{\text{minimize}}\ \big(-f(\boldsymbol{x})\big)},$$

$$\underbrace{\underset{\boldsymbol{x}}{\text{minimize}}\ f(\boldsymbol{x})} = \underbrace{-\underset{\boldsymbol{x}}{\text{maximize}}\ \big(-f(\boldsymbol{x})\big)}.$$

We can have similar conversions for the arguments of maximization and minimization but as the sign of optimal value of function is not important in argument, we do not have the negative sign before maximization and minimization:

$$\arg\max_{\boldsymbol{x}} f(\boldsymbol{x}) = \overbrace{\arg\min_{\boldsymbol{x}} \big(-f(\boldsymbol{x})\big)},$$

$$\arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \underbrace{\arg\max_{\boldsymbol{x}} \big(-f(\boldsymbol{x})\big)}.$$



Converting optimization problems

Converting max to min and vice versa

We can convert convert maximization to minimization and vice versa using the reciprocal of cost function:

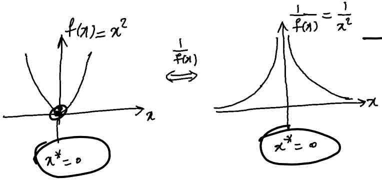
$$\star \quad \underset{\boldsymbol{x}}{\text{maximize}} \;\; f(\boldsymbol{x}) = \frac{1}{\underset{\boldsymbol{x}}{\text{minimize}} \;\; \frac{1}{f(\boldsymbol{x})}},$$

$$\underset{\boldsymbol{x}}{\text{minimize}} \;\; f(\boldsymbol{x}) = \frac{1}{\underset{\boldsymbol{x}}{\text{maximize}} \;\; \frac{1}{f(\boldsymbol{x})}}.$$

We can have similar conversions for the arguments of maximization and minimization:

$$\arg\max_{\boldsymbol{x}} f(\boldsymbol{x}) = \overbrace{\arg\min_{\boldsymbol{x}} \frac{1}{f(\boldsymbol{x})}},$$

$$\arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \arg\max_{\boldsymbol{x}} \frac{1}{f(\boldsymbol{x})}.$$

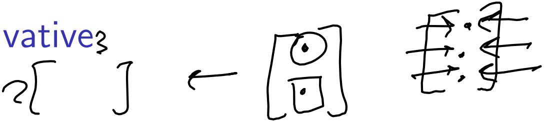


$\rightarrow \; \text{max} = \infty \; \rightarrow \; \frac{1}{\infty} = 0$

$\text{argmax} = 0$

Preliminaries on Derivatives

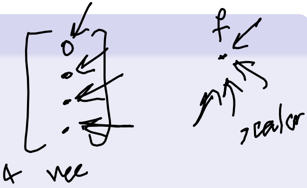
Dimensionality of derivatives



- Consider a function $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$.
- Derivative of function $f(\boldsymbol{x}) \in \mathbb{R}^{d_2}$ with respect to (w.r.t.) $\boldsymbol{x} \in \mathbb{R}^{d_1}$ has dimensionality $(d_1 \times d_2)$.
- This is because tweaking every element of $\boldsymbol{x} \in \mathbb{R}^{d_1}$ can change every element of $f(\boldsymbol{x}) \in \mathbb{R}^{d_2}$. The (i, j)-th element of the $(d_1 \times d_2)$-dimensional derivative states the amount of change in the j-th element of $f(\boldsymbol{x})$ resulted by changing the i-th element of $\boldsymbol{x}$.

Examples

- The derivative of a scalar w.r.t. a scalar is a scalar.
- The derivative of a scalar w.r.t. a vector is a vector.
- The derivative of a scalar w.r.t. a matrix is a matrix.
- The derivative of a vector w.r.t. a vector is a matrix.
- The derivative of a vector w.r.t. a matrix is a rank-3 tensor.
- The derivative of a matrix w.r.t. a matrix is a rank-4 tensor.



Dimensionality of derivative

$A \rightarrow 3 \times \boxed{2}$ $\quad B \rightarrow \boxed{2} \times O$

In more details:

- If the function is $f : \mathbb{R} \to \mathbb{R}, f : x \mapsto f(x)$, the derivative $(\partial f(x)/\partial x) \in \mathbb{R}$ is a scalar because changing the scalar $\boldsymbol{x}$ can change the scalar $f(\boldsymbol{x})$.
- If the function is $f : \mathbb{R}^d \to \mathbb{R}, f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, the derivative $(\partial f(\boldsymbol{x})/\partial \boldsymbol{x}) \in \mathbb{R}^d$ is a vector because changing every element of the vector $\boldsymbol{x}$ can change the scalar $f(\boldsymbol{x})$.
- If the function is $f : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}, f : \boldsymbol{X} \mapsto f(\boldsymbol{X})$, the derivative $(\partial f(\boldsymbol{X})/\partial \boldsymbol{X}) \in \mathbb{R}^{d_1 \times d_2}$ is a matrix because changing every element of the matrix $\boldsymbol{X}$ can change the scalar $f(\boldsymbol{X})$.
- If the function is $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}, f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, the derivative $(\partial f(\boldsymbol{x})/\partial \boldsymbol{x}) \in \mathbb{R}^{d_1 \times d_2}$ is a matrix because changing every element of the vector $\boldsymbol{x}$ can change every element of the vector $f(\boldsymbol{x})$.
- If the function is $f : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_3}, f : \boldsymbol{X} \mapsto f(\boldsymbol{X})$, the derivative $(\partial f(\boldsymbol{X})/\partial \boldsymbol{X})$ is a $(d_1 \times d_2 \times d_3)$-dimensional tensor because changing every element of the matrix $\boldsymbol{X}$ can change every element of the vector $f(\boldsymbol{X})$.
- If the function is $f : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_3 \times d_4}, f : \boldsymbol{X} \mapsto f(\boldsymbol{X})$, the derivative $(\partial f(\boldsymbol{X})/\partial \boldsymbol{X})$ is a $(d_1 \times d_2 \times d_3 \times d_4)$-dimensional tensor because changing every element of the matrix $\boldsymbol{X}$ can change every element of the matrix $f(\boldsymbol{X})$.

Gradient, Jacobian, and Hessian

$\triangleq$ $\boxed{:=}$ $\boxed{\leftarrow}$

Definition (Gradient)

Consider a $\underline{\text{function } f : \mathbb{R}^d \boxed{\to} \mathbb{R}}$, $\underline{f : \boldsymbol{x} \boxed{\mapsto} f(\boldsymbol{x})}$. In optimizing the function f, the derivative of function w.r.t. its variable $\boldsymbol{x}$ is called the **gradient**, denoted by:

$$\underbrace{\nabla f(\boldsymbol{x})}_{} \overset{\text{def}\leftarrow}{\boxed{:=}} \underbrace{\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}}_{} \in \boxed{\mathbb{R}^d.}$$

Definition (Hessian)

Consider a function $f : \mathbb{R}^d \to \mathbb{R}$, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$. The second derivative of function w.r.t. to its derivative is called the **Hessian** matrix, denoted by:

$B_{ij} = B_{ji}$

$B = B^T$

$$\boldsymbol{B} = \underbrace{\nabla^2 f(\boldsymbol{x})}_{} := \underbrace{\frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}^2}}_{} \in \boxed{\mathbb{R}^{d\times d}.}$$

The Hessian matrix is <u>symmetric</u>. If the function is <u>convex</u>, its Hessian matrix is <u>positive semi-definite.</u>

$$\frac{\partial f}{\partial x} \in \mathbb{R}^d = g \rightarrow \frac{\partial \boxed{g} \rightarrow \mathbb{R}^d}{\partial \boxed{x} \rightarrow \mathbb{R}^d} \in \mathbb{R}^{d\times d}$$

Gradient, Jacobian, and Hessian

$$\begin{bmatrix} x_1 \\ \vdots \\ x_{d_1} \end{bmatrix} \qquad \begin{bmatrix} \boxed{f_1} \\ f_2 \\ \vdots \\ f_{d_2} \end{bmatrix}$$

Definition (Jacobian)

If the function is multi-dimensional, i.e., $\underline{f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}}$, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, the gradient becomes a matrix:

$$\boldsymbol{J} := \left[\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_{d_1}}\right]^\top = \underbrace{\begin{bmatrix} \boxed{\frac{\partial f_1}{\partial x_1}} & \dots & \boxed{\frac{\partial f_{d_2}}{\partial x_1}} \\ \vdots & \ddots & \vdots \\ \boxed{\frac{\partial f_1}{\partial x_{d_1}}} & \dots & \boxed{\frac{\partial f_{d_2}}{\partial x_{d_1}}} \end{bmatrix}} \in \mathbb{R}^{\underline{d_1 \times d_2}},$$

where $\boldsymbol{x} = [x_1, \dots, x_{d_1}]^\top$ and $f(\boldsymbol{x}) = [f_1, \dots, f_{d_2}]^\top$.

This matrix derivative is called the **Jacobian** matrix.

Optimization Problems

General optimization problem

Consider the function $f : \mathbb{R}^d \to \mathbb{R}$, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$. Let the domain of function be $\mathcal{D}$ where $\boldsymbol{x} \in \mathcal{D}, \boldsymbol{x} \in \mathbb{R}^d$.

Definition (Unconstrained optimization)

Unconstrained minimization of a cost function $f(.)$:

$$\boxed{\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}),}$$

where $\boldsymbol{x}$ is called the **optimization variable** and the function $f(.)$ is called the **objective function** or the **cost function**.

General optimization problem

Definition (Constrained optimization)

Constrained optimization problem where we want to minimize the function $f(\boldsymbol{x})$ while satisfying m_1 inequality constraints and m_2 equality constraint:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} && f(\boldsymbol{x}) \\ &\text{subject to} && y_i(\boldsymbol{x}) \leq 0, \quad i \in \{1, \ldots, m_1\}, \\ & && h_i(\boldsymbol{x}) = 0, \quad i \in \{1, \ldots, m_2\}. \end{aligned}$$

$$\begin{aligned} &y_1(x) \leq 0 \\ &y_2(x) \leq 0 \\ &\vdots \\ &y_{m_1}(x) \leq 0 \end{aligned}$$

$f(\boldsymbol{x})$ is the **objective function**, every $y_i(\boldsymbol{x}) \leq 0$ is an **inequality constraint**, and every $h_i(\boldsymbol{x}) = 0$ is an **equality constraint**.

$$\begin{aligned} &h_1(x) = 0 \\ &\vdots \\ &h_{m_2}(x) = 0 \end{aligned}$$

Note

If some of the inequality constraints are not in the form $y_i(\boldsymbol{x}) \leq 0$, we can restate them as:

$$\begin{aligned} y_i(\boldsymbol{x}) \geq 0 &\Longrightarrow -y_i(\boldsymbol{x}) \leq 0, \\ y_i(\boldsymbol{x}) \leq c &\Longrightarrow \underbrace{y_i(\boldsymbol{x}) - c}_{g(x)} \leq 0. \end{aligned}$$

Therefore, all inequality constraints can be written in the form $y_i(\boldsymbol{x}) \leq 0$.

General optimization problem

$$\underset{x}{\text{minimize}} \qquad \underset{\{x_1,x_2\}}{\text{minimize}}$$

Example:

$$
[x_1,x_2]^T \leftarrow
\begin{array}{ll}
\underset{\boxed{x}}{\text{minimize}} & \boxed{x_1 + 3x_2^2} \\
\text{subject to} & \left.\begin{array}{l} 2x_1 - 10x_2 \leq \boxed{5}, \\ -2x_1 + 5x_2 \boxed{\geq} 3, \\ \underbrace{4x_1 + 10x_2 = 6.} \end{array}\right\}
\end{array}
$$

can be converted to:

$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & x_1 + 3x_2^2 \\
\text{subject to} & \left.\begin{array}{l} 2x_1 - 10x_2 \boxed{-5} \leq 0, \\ \underbrace{2x_1 - 5x_2 + 3 \leq 0,} \end{array}\right\} \rightarrow m_1 = 2 \\
 & 4x_1 + 10x_2 - 6 = 0. \longrightarrow m_2 = 1
\end{array}
$$

Feasible point

Definition (Feasible point)

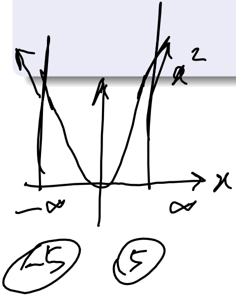
$x \geq 0$

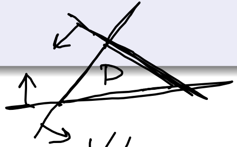
The point $\boldsymbol{x}$ for the optimization problem:

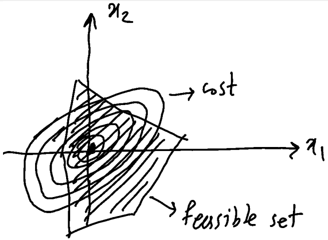
$$
\star \left\{
\begin{array}{ll}
\underset{\boldsymbol{x}}{\text{minimize}} & \boxed{f(\boldsymbol{x})} \\
\text{subject to} & \underline{y_i(\boldsymbol{x}) \leq 0}, \; i \in \{1, \ldots, m_1\}, \\
 & \underline{h_i(\boldsymbol{x}) = 0}, \; i \in \{1, \ldots, m_2\},
\end{array}
\right.
$$

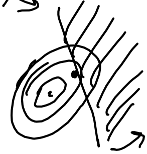
is feasible if:

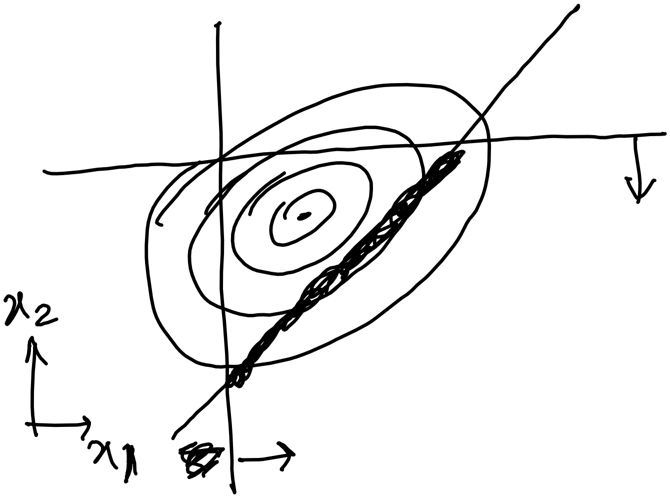
$$
\begin{array}{l}
\boxed{\boldsymbol{x} \in \mathcal{D},} \text{ and} \\
y_i(\boldsymbol{x}) \leq 0, \quad \forall i \in \{1, \ldots, m_1\}, \text{ and} \\
h_i(\boldsymbol{x}) = 0, \quad \forall i \in \{1, \ldots, m_2\}.
\end{array}
$$

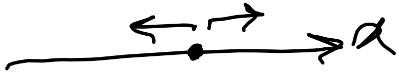












$x \in \mathbb{R}$

line → dot

$x \in \mathbb{R}^2$

line → line

$x \in \mathbb{R}^3$

line → plane

hyperplane

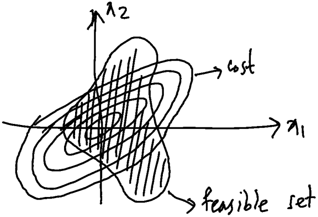
Constrained optimization with the feasible set

Definition (Constrained optimization)

The **constrained** optimization problem can also be stated as:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & f(\boldsymbol{x}) \\ \text{subject to} & \underbrace{\boldsymbol{x} \in \boxed{\mathcal{S},}} \end{array}$$

where $\mathcal{S}$ is the feasible set of constraints.



Converting Constrained Optimization to Unconstrained Optimization

Converting Constrained Optimization to Unconstrained Optimization

- Consider the following constrained optimization problem:

min $f(x)$

→ min $g(x)$

☆
$$\begin{cases} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \boxed{g(\mathbf{x}) \leq 0.} \end{cases} \qquad (7)$$

→ 0.001

This is a **constrained hard penalty** because it kills the problem if it is not satisfied (it can never happen as it would be **infeasible**).

- We can convert it to an unconstrained regularized problem:

≃ ☆

$$\underset{\mathbf{x}}{\text{minimize}} \quad \underbrace{\underbrace{f(\mathbf{x})} + \boxed{\lambda}\, \underbrace{g(\mathbf{x})}}_{\text{regularized cost}}, \qquad (8)$$

→ regularization parameter

where $\lambda > 0$ is the regularization parameter. This is a **soft penalty** which tolerates some violence of the constraint. The solution of this problem is approximately similar to the solution of the constrained hard penalty.

0.001

0.1

$$\left\{\begin{array}{l}\min\limits_{x} \; f(x) \\ \text{s.t.} \quad y_1(x) \le 0 \\ \qquad\;\; y_2(x) \le 0 \\ \qquad\;\; h(x) = 0 \end{array}\right\}$$

$$\min_{x,\lambda_1,\lambda_2,\nu} \mathcal{L} = \underbrace{f(x) + \lambda_1 y_1(x) + \lambda_2 y_2(x) + \nu h(x)}_{\text{Lagrangian}}$$

$\lambda_1, \lambda_2, \nu$:

- Lagrange multiplier
- dual variables
- regularization params

$$\frac{\partial \mathcal{L}}{\partial x} \overset{\text{set}}{=} 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_1} \overset{\text{set}}{=} 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_2} \overset{\text{set}}{=} 0$$

Converting Constrained Optimization to Unconstrained Optimization

$$\lim_{\lambda\to\infty} f(x) + \underbrace{\boxed{\lambda g(x)}}_{\lambda\ \boxed{\star}} \qquad \underbrace{g(x) \leq 0}$$

- We can also convert it to an **unconstrained** problem using indicator function. In optimization, **indicator function** $\mathbb{I}(.)$ is zero if its condition is satisfied and is infinite otherwise.

$$\mathbb{I}(\underbrace{\overset{g(x)\leq 0}{\boxed{\boldsymbol{x} \in \mathcal{S}}}}) = \begin{cases} \underline{0} & \text{if } \boldsymbol{x} \in \mathcal{S} \\ \underline{\infty} & \text{if } \boldsymbol{x} \notin \mathcal{S}. \end{cases} \tag{9}$$

The problem can become regularized using the indicator function:

$$\star \longrightarrow \quad \underset{\boldsymbol{x}}{\text{minimize}} \quad \underbrace{f(\boldsymbol{x}) + \overbrace{\underbrace{\lambda\mathbb{I}(g(\boldsymbol{x}) \leq 0)}_{\infty}}^{=0\ \to\ \infty}}, \tag{\underline{10}}$$

where $\lambda > 0$ is the regularization parameter. This is a **unconstrained hard penalty** because it kills the problem if it is not satisfied (it blows the regularized cost function to infinity). The solution of this problem is exactly the same as the solution of the constrained hard penalty.

Converting Constrained Optimization to Unconstrained Optimization

- In practice, it is not possible to implement the indicator function in computer as it has infinity and also it is not smooth (differentiable). Therefore, we can use approximations of the indicator function by **barrier functions**. One of the barrier functions is logarithm, named the **logarithmic barrier** or **log barrier** in short. It approximates the indicator function by:

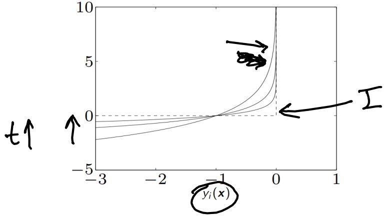
Newton's method

log-barrier method

+

$$\underbrace{\mathbb{I}(y_i(\boldsymbol{x}) \leq 0)}\boxed{\approx}\overbrace{\underbrace{-\frac{1}{t}\log(-y_i(\boldsymbol{x})),}} \quad ★ \qquad (11)$$

where $t > 0$ (usually a large number such as $t = 10^6$) and the approximation becomes more accurate by $t \to \infty$.



The solution of the problem, which uses barrier functions as approximations of the indicator function, is **approximately** similar to the solution of the hard penalty. This is the technique that the **interior point method** uses.

References

[1] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "KKT conditions, first-order and second-order optimization, and distributed optimization: Tutorial and survey," *arXiv preprint arXiv:2110.01858*, 2021.

[2] Y. Nesterov, *Lectures on convex optimization*, vol. 137.
Springer, 2018.