Optimization for machine learning

Introduction to numerical optimization

R. Flamary

March 5, 2020

Objective of this course

Optimization in machine learning and data science

- All ML and data science methods rely on numerical optimization.
- Understanding the method $\equiv$ understanding the optimization problem.
- What is inside the black box of the skikit-learn `.fit()` function ?

Your objective

- Recognize the properties of optimization problems.
- Understand the optimization problems in ML approaches.
- Model new optimization problems (new ML method).
- Find a proper algorithm for a given problem.
- Be able to implement an optimization algorithm.

Full course overview

1. **Introduction to numerical optimization**
   - 1.1 Optimization problem formulation and principles
   - 1.2 Properties of optimization problems
   - 1.3 Machine learning as an optimization problem
2. **Constrained Optimization and Standard Optimization problems**
   - 2.1 Constraints, Lagrangian and KKT
   - 2.2 Linear Program (LP)
   - 2.3 Quadratic Program (QP)
   - 2.4 Other Classical problems (MIP,QCQP,SOCP,SDP)
3. **Smooth Optimization**
   - 3.1 Gradient descent
   - 3.2 Newton, quasi-Newton and Limited memory
   - 3.3 Stochastic Gradient Descent
4. **Non-smooth Optimization**
   - 4.1 Proximal operator and proximal methods
   - 4.2 Conditional gradient
5. **Conclusion**
   - 5.1 Other approaches (Coordinate descent, DC programming)
   - 5.2 Optimization problem decision tree
   - 5.3 References an toolboxes

Course overview

Numerical optimization problem

Problem formulation

$$\min_{\mathbf{x}\in\mathcal{C}} \quad F(\mathbf{x}) \tag{1}$$

- F is the objective function (sometime called cost function).
- $\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ is a vector of n variables.
- $\mathcal{C} \subseteq \mathbb{R}^n$ is the set of admissible solutions.
- Objective : Find a solution $\mathbf{x}^\star \in \mathcal{C}$, having the minimal value for F such that

$$F(\mathbf{x}^\star) \leq F(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{C}.$$

Assumptions (in this course)

- The problem is proper (there exists a solution), F is lower bounded on $\mathcal{C}$.
- You have access to F and $\mathcal{C}$ (mathematical expression, no black box).

Notation : Lowercase bold is a vector, Uppercase bold is a matrix.

Standard constrained optimization

Problem reformulation

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & F(\mathbf{x}) \\ \text{with} & h_j(\mathbf{x}) = 0 \quad \forall j = 1, \dots, p \\ \text{and} & g_i(\mathbf{x}) \leq 0 \quad \forall i = 1, \dots, q. \end{array} \tag{2}$$

- This problem is equivalent to (7) when $\mathcal{C}$ can be expressed as

$$\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{R}^n \;\middle|\; h_j(x) = 0,\ \forall j = 1, \dots, p \text{ and } g_i(x) \leq 0,\ \forall i = 1, \dots, q \right\}.$$

- h_j and g_i define respectively the equality and inequality constraints.
- When $p = q = 0$ the problem is said to be unconstrained and $\mathcal{C} = \mathbb{R}^n$.
- The complexity of solving problems (7) and (2) depends on the properties of F and $\mathcal{C}$.
- Problem above is a standard formulation for constrained optimization.

Definitions

$$\min_{\mathbf{x}\in\mathcal{C}} \quad F(\mathbf{x})$$

Feasible point

Any point $\mathbf{x} \in \mathcal{C}$ that satisfy the constraints in set $\mathcal{C}$.

Optimal value

Minimal value function on the feasible set $\mathcal{C}$, often denoted as $F^\star$.

Optimality/Optimal solution

$\mathbf{x}^\star \in \mathcal{C}$ is a solution of the optimization problem if satisfies the constraints in set $\mathcal{C}$ and

$$F(\mathbf{x}^\star) \leq F(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{C}.$$

$\mathbf{x}^\star$ might not be unique in the general case.

Sub-optimal point

$\mathbf{x} \in \mathcal{C}$. is an ϵ-suboptimal point of the problem for $\epsilon > 0$ if

$$F(\mathbf{x}) \leq F(\mathbf{x}^\star) + \epsilon$$

Active constraint

g_i is considered an active constraint in $\mathbf{x}$ if $g_i(\mathbf{x}) = 0$.

Exercise 1: Positive least square reformulation

Problem

$$\min_{\mathbf{x}\geq\mathbf{0}} \quad \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \qquad \text{with } \mathbf{y} \in \mathbb{R}^m \text{ and } \mathbf{H} \in \mathbb{R}^{m\times n}$$

Exercise

1. Express $F(\mathbf{x})$ and $\mathcal{C}$ for the problem above.
2. Find p, q the number of constraints :
3. Express h_j and g_i if there are somme constraints:

Numerical optimization algorithm

$$\min_{\mathbf{x}\in\mathcal{C}} \quad F(\mathbf{x})$$

Iterative optimization algorithm

An iterative algorithm A is an algorithm providing a series $\mathbf{x}^{(k)}$ for $k = 0, 1, \ldots$ of iterates $\mathbf{x}^{(k+1)} = A(\mathbf{x}^{(k)})$ that converges to a solution $\mathbf{x}^\star$ of the optimization problem starting from an initial guess $\mathbf{x}^{(0)}$.

- If $F(\mathbf{x}^{(k+1)}) \leq F(\mathbf{x}^{(k)})$, $\forall k$ then it is called a **descent algorithm**.
- In practice iterations are stopped when a convergence criterion is met.

Convergence of iterative methods

- The convergence speed can be expressed in objective value

$$|F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^\star)| \leq \gamma |F(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^\star)|^q \tag{3}$$

- Or it can be expressed in terms of iterates:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\| \leq \gamma \|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\|^q \tag{4}$$

where $\gamma \in [0, 1)$ and $q \geq 1$ is the convergence order ($q = 1$ linear, $q = 2$ quadratic...).

Properties of optimization problems

Know your optimization problem (and its properties)

- They with guide you toward the proper solver.
- They tell you how much you can trust the solution (well posed, unique solution).
- They will help you design the optimization problem.

Convexity

- Well posed problem.
- Unique solution when strict convexity.

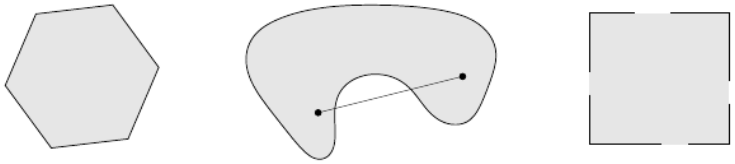
Smoothness

- Continuity, differentiability
- When function smooth, one can use its gradients.

Solutions

- What is a solution of the optimization problem ?
- Criterions for reaching a solution (stopping the algorithm).

Convex set



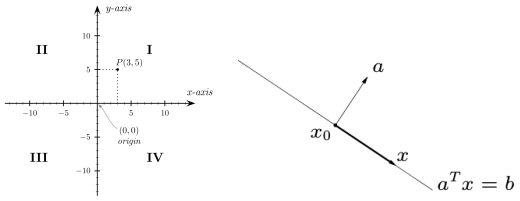
Definition: Convex set

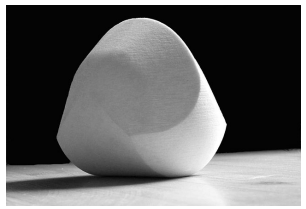
$\mathcal{C} \subset \mathbb{R}^n$ is a convex set if for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{C}^2$ and for any $0 \leq \alpha \leq 1$ we have

$$\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in \mathcal{C}$$

Image from [Boyd and Vandenberghe, 2004]

Examples of convex sets

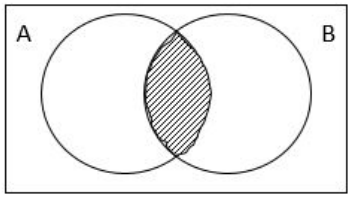


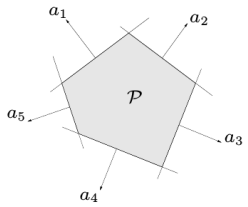


Examples

- $\mathbb{R}^n$
- Positive orthant of $\mathbb{R}^n$: $\mathbb{R}^n_+$.
- Hyperplan : $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{a}^\top \mathbf{x} = b\}$
- Half space: $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{a}^\top \mathbf{x} \leq b\}$
- Polyhedra: $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$
- Gömböc

Operations on set preserving convexity (1)





Intersection

If $\mathcal{X}_k$ are convex set $\forall k$ then their intersection

$$\bigcap_{k=1}^{K} \mathcal{X}_k$$

is also convex.

Operations on set preserving convexity (2)

Cartesian product

If $\mathcal{X}_k \subset \mathbb{R}^{n_k}$, are convex $\forall k = 1, \cdots, M$ then

$$\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \times \mathcal{X}_M = \{(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M) : \mathbf{x}_k \in \mathcal{X}_k\}$$

is convex.

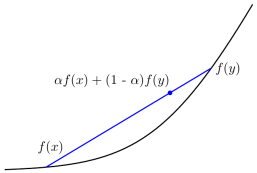
Affine transform

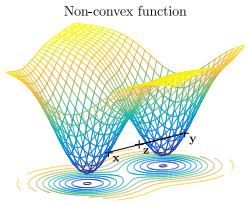
If $\mathcal{X} \subset \mathbb{R}^d$ is convex and $\mathcal{A}(\mathbf{x}) \mapsto \mathbf{A}\mathbf{x} + \mathbf{b}$ is an affine transform defined by matrix $\mathbf{A} \in \mathbb{R}^{p \times d}$ and vector $\mathbf{b}$ then

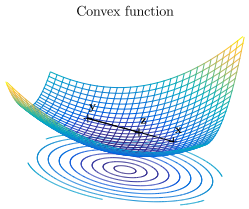
$$\mathcal{A}(\mathcal{X}) = \{\mathcal{A}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$$

is convex. These transformations include translation and rotations.

Convex function







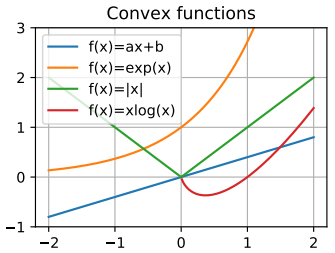
Definition: Convex function

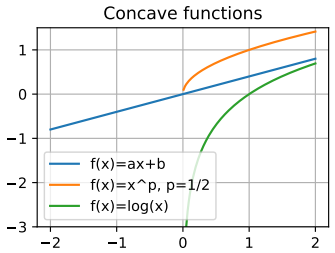
A function F is said to be convex if it lies below its chords, that is

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \quad F(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha F(\mathbf{x}) + (1-\alpha)F(\mathbf{y}), \text{ with } 0 \leq \alpha \leq 1. \quad (5)$$

- A function is said to be strictly convex when the two inequalities are strict.
- Strict convexity implies that the function has a unique minimum.
- If a function F is convex, then the set $\{\mathbf{x} \in \mathbb{R}^n \mid F(\mathbf{x}) \leq 0\}$ is convex.
- A function F is concave if $-F$ is convex.

Examples of functions in $\mathbb{R}$





Convex functions

- Affine functions : $x \mapsto ax + b$ pour tout $a, b \in \mathbb{R}$.
- Exponential functions : $x \mapsto e^{ax}$ pour tout $a \in \mathbb{R}$.
- Power of absolute value : $x \mapsto |x|^p$, pour tout $p \geq 1$.
- Neg-entropy : $x \mapsto x \log x$ pour $x > 0$

Concave Functions

- Affine functions : $x \mapsto ax + b$ pour tout $a, b \in \mathbb{R}$.
- Power : $x \mapsto x^p$, pour $x > 0$ et pour tout $0 \leq p \leq 1$.
- Logarithm : $x \mapsto \log x$ pour $x > 0$

Operations preserving convexity (1)

Positive sum

Let $\lambda_1, \lambda_2 \geq 0$ and f_1, f_2 two convex function then

$$\lambda_1 f_1 + \lambda_2 f_2$$

is convex.

Composition with affine function

let $\mathbf{A} \in \mathbb{R}^{p \times d}$ and $b \in \mathbb{R}^p$ and $f : \mathbb{R}^p \mapsto \mathbb{R}$ be a convex function, the the composition

$$f(\mathbf{A}\mathbf{x} + b)$$

is convex

Example

- Log barrier : $f(\mathbf{x}) = -\sum_{i=1}^{m} \log(b_i - \mathbf{a}_i^\top \mathbf{x})$ with dom$f = \{\mathbf{x} : \mathbf{a}_i^\top \mathbf{x} \leq b_i\}$
- Norm of an affine function : $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} + b\|$

Operations preserving convexity (2)

Composition

- let $g : \mathbb{R}^d \mapsto \mathbb{R}$ be a convex function and $h : \mathbb{R} \mapsto \mathbb{R}$ be a convex and increasing function, then
$$f(\mathbf{x}) = h(g(\mathbf{x}))$$
is convex.

Maximum

- If $f_1, \cdots, f_m$ are convex functions then
$$f(\mathbf{x}) = \max_i \{f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})\}$$
is convex.

Example

- Piecewise linear function: $f(\mathbf{x}) = \max_{i=1,\cdots,m}(\mathbf{a}_i^\top \mathbf{x} + b)$

Convexity in optimization

Convex optimization problem

$$\min_{\mathbf{x} \in \mathcal{C}} \quad F(\mathbf{x})$$

- The problem is convex if F is a convex function and $\mathcal{C}$ is a convex set.
- Any local minimizer of a convex function is a global minimizer.
- If the function is strictly convex the minimizer is unique.
- Maximizing a concave function under convex constraints is a convex problem.

Disciplined Convex Programming [Grant et al., 2006]

- Express the objective function and constraints as combination and composition of operations preserving convexity.
- Allows for designing generic solvers (Matlab [Grant and Boyd, 2014], Python [Diamond and Boyd, 2016]).

Smoothness and continuity

Differentiability classes

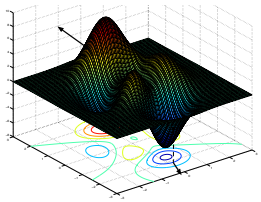
Let f be a real function. Then f is of differentiability class C^k if and only if $\frac{d^k f(x)}{dx^k}$ is continuous.

- C^0 is the set of continuous real functions.
- C^1 is the set of real functions with continuous derivatives.
- A real function f is **smooth** if it is of differentiability class C^∞.

Exercise 2: Differentiability and convexity

Function	Diff. Class	Convexity
$f(x) = x^2$		
$f(x) = e^x$		
$f(x) = \|x\|$		
$f(x) = \max(x, 0)$		
$f(x) = \text{sign}(x)$		
$f(x) = \log(1 + \exp(x))$		
$f(x) = 2x + 1$		
$f(x) = \max(x, 0)^2$		

Gradient of a function



Gradient
The gradient $\boldsymbol{\nabla} F(\mathbf{x})$ of a function $F : \mathbb{R}^n \to \mathbb{R}$ at point $\mathbf{x}$ is the vector whose components are the partial derivatives of F

$$\nabla_{\mathbf{x}} F(\mathbf{x}) = \left[\frac{\partial F(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial F(\mathbf{x})}{\partial x_n} \right]^T \tag{6}$$

- If the gradient exists $\forall \mathbf{x}$ in the domain of F, the function F is called **differentiable**.
- $\nabla_{\mathbf{x}} F(\mathbf{x})$ give the steepest direction (where F is increasing the most).
- The vector normal to surface $(\mathbf{x}, F(\mathbf{x}))$ is given by $(\nabla_{\mathbf{x}} F(\mathbf{x}), -1)$.

Exercise 3: Gradient computation

Two variables

$$F(\mathbf{x}) = x_1 - x_1 x_2 - x_2$$

Compute the gradient $\nabla_{\mathbf{x}} F(\mathbf{x})$:

$$\nabla_{\mathbf{x}} F(\mathbf{x}) =$$

Quadratic loss

$$F(\mathbf{x}) = \|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2$$

Compute the gradient $\nabla_{\mathbf{x}} F(\mathbf{x})$:

$$\nabla_{\mathbf{x}} F(\mathbf{x}) =$$

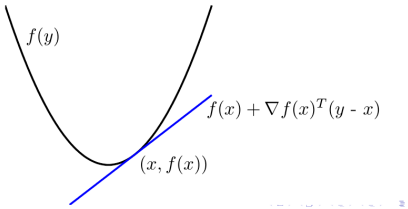
Exponential with linear function

$$F(\mathbf{x}) = \exp(\mathbf{w}^T \mathbf{x} + b)$$

Compute the gradient $\nabla_{\mathbf{x}} F(\mathbf{x})$:

$$\nabla_{\mathbf{x}} F(\mathbf{x}) =$$

Gradient and convexity



Gradient of a convex function
F a differentiable function is **convex** if and only if

$$F(\mathbf{y}) \geq F(\mathbf{x}) + \nabla F(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \quad \forall \mathbf{y}, \mathbf{x} \in \text{dom} F \tag{7}$$

- A convex function is lower bounded by its local linear approximation.
- For unconstrained problems with $\mathcal{C} = \mathbb{R}^n$, if F is convex and differentiable, $\mathbf{x}$ is a global minimum if and only if

$$\nabla_{\mathbf{x}} F(\mathbf{b}) = \mathbf{0}$$

Hessian and second derivatives

Hessian of a function
The Hessian matrix $\mathbf{H} = \nabla^2_{\mathbf{x}} F(\mathbf{x})$ of a twice differentiable function F is the matrix whose components can be expressed as

$$H_{i,j} = \left(\nabla^2_{\mathbf{x}} F(\mathbf{x})\right)_{i,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_j}$$

- F is a convex function if and only if $\nabla^2_{\mathbf{x}} F(\mathbf{x})$ is semi definite positive $\forall \mathbf{x} \in \text{dom} F$.
- If $\nabla^2_{\mathbf{x}} F(\mathbf{x})$ is strictly positive definite $\forall \mathbf{x} \in \text{dom} F$ then F is strictly convex.
- The order 2 Taylor approximation of the function around $\mathbf{x}_0$ can be expressed as

$$F(\mathbf{x}) \approx F(\mathbf{x}_0) + \underbrace{\nabla_{\mathbf{x}_0} F(\mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_0)}_{\text{Linear term}} + \underbrace{(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H} (\mathbf{x} - \mathbf{x}_0)}_{\text{Quadratic term}} \tag{8}$$

The approximation is exact if F is a polynomial of order ≤ 2

Exercise 4: Hessian computation

Two variables

$$F(\mathbf{x}) = x_1 - x_1 x_2 - x_2$$

Compute the Hessian $\nabla^2_{\mathbf{x}} F(\mathbf{x})$, is is positive semi definite ?

$$\nabla^2_{\mathbf{x}} F(\mathbf{x}) =$$

Quadratic loss

$$F(\mathbf{x}) = \|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2$$

Compute the Hessian $\nabla^2_{\mathbf{x}} F(\mathbf{x})$, is is positive semi definite ?

$$\nabla^2_{\mathbf{x}} F(\mathbf{x}) =$$

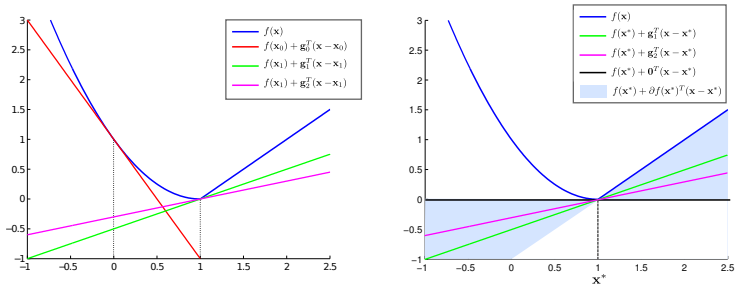
Exponential with linear function

$$F(\mathbf{x}) = \exp(\mathbf{w}^T \mathbf{x} + b)$$

Compute the Hessian $\nabla^2_{\mathbf{x}} F(\mathbf{x})$, is is positive semi definite ?

$$\nabla^2_{\mathbf{x}} F(\mathbf{x}) =$$

Subgradients and subdifferential



Non differentiable function

- For a convex function $F(\mathbf{x})$, $\mathbf{g}$ is a subgradient of F in $\mathbf{x}_0$ if

$$F(\mathbf{x}) \geq F(\mathbf{x}_0) + \mathbf{g}^\top(\mathbf{x} - \mathbf{x}_0) \tag{9}$$

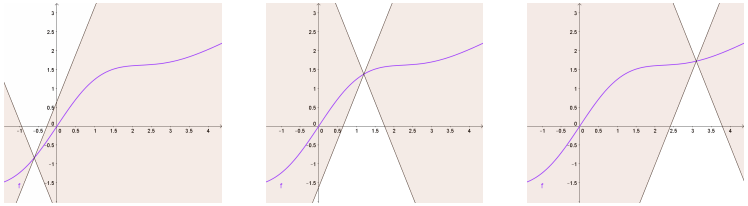
- The set of all subgradients at $\mathbf{x}_0$ is the subdifferential $\partial f(\mathbf{x}_0)$.
- If F is differentiable in $\mathbf{x}_0$ there is a unique subgradient: $\partial f(\mathbf{x}_0) = \{\nabla_{\mathbf{x}} F(\mathbf{x})\}$
- $\mathbf{x}^\star$ is a minimum of the unconstrained convex function F if $\mathbf{0} \in \partial F(\mathbf{x}^\star)$.

Exercise 5: Subgradients

Find the subdifferential $\partial F(\mathbf{x})$ for the following functions:

1. $F(x) = |x|$, at $x \in \{-1, 0, 1\}$
2. $F(x) = \max(x, 0)$, at $x \in \{-1, 0, 1\}$
3. $F(x) = \max(x, 0) + x$, at $x \in \{-1, 0, 1\}$
4. $F(x) = |x| + x^2$, at $x \in \{-1, 0, 1\}$

Lipschitz continuity



Lipschitz function

Function F is called **Lipschitz** or **Lipschitz continuous** if there exists a constant $K > 0$ such that $\forall \mathbf{x}, \mathbf{y} \in \mathcal{C}^2$

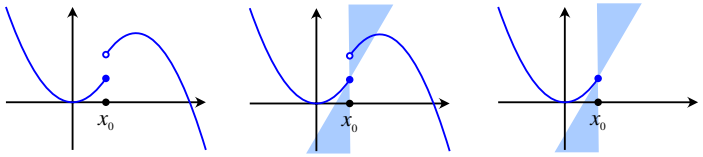
$$|F(\mathbf{x}) - F(\mathbf{y})| \leq K\|\mathbf{x} - \mathbf{y}\| \tag{10}$$

- A K satisfying the above constraint is called a Lipschitz constant of the function.
- If $K < 1$ the function is a contraction.
- Function F is **gradient Lipschitz** if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{C}^2$

$$\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq K\|\mathbf{x} - \mathbf{y}\| \tag{11}$$

- Lipschitz functions can be easily upper bounded,

Semicontinuity



Lower semi-continuous function

A function F is **lower semi-continuous (l.s.c.)** if for any point $\mathbf{x}_0 \in \mathcal{C}$ we have

$$F(\mathbf{x}_0) \leq \lim_{\mathbf{x} \to \mathbf{x}_0} \inf F(\mathbf{x}) \tag{12}$$

- Continuous functions are l.s.c. since it implies the equality above.
- If the function is l.s.c., there exists a local affine minorant.
- If the function is l.s.c. and convex it means that the sub-differential is never empty and the minorant is global.

Constraints VS non-smooth

Characteristic function

Let A be a subset of $\mathbb{R}^n$, the **characteristic function** χ_A of A is the function

$$\chi_A(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \in A \\ +\infty, & \text{if } \mathbf{x} \notin A \end{cases} \tag{13}$$

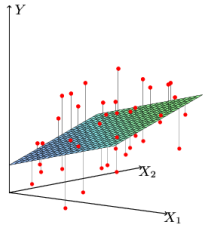
- If A is a closed set, χ_A is lower semi-continuous.
- If A is a closed convex set, χ_A is convex.

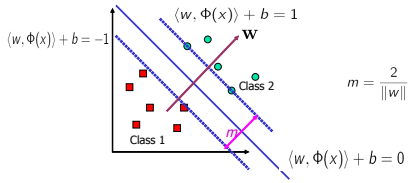
Equivalent optimization problems

$$\min_{\mathbf{x} \in \mathcal{C}} F(\mathbf{x}) \quad \equiv \quad \min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) + \chi_{\mathcal{C}}(\mathbf{x})$$

- Constrained OP can be reformulated as a non-smooth unconstrained OP.
- The new objective function is a sum of two functions (splitting algorithms).

Convexity and smoothness in machine learning





Convex and smooth problems

- Smooth problem provides us with gradients for iterative methods.
- Convexity means the a solution of the problem is global.
- Convexity leads to several efficient algorithms.

ML approaches relying on convex problems

- Least square regression, Lasso.
- Support Vector Machines.
- Logistic and multinomial regression.

Local and global solutions

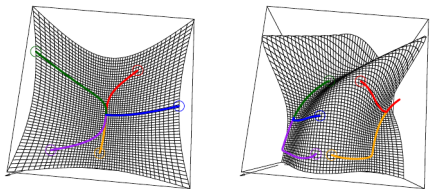
$$\min_{\mathbf{x} \in \mathcal{C}} F(\mathbf{x})$$

Local solution

For the optimization problem above, a feasible point $\mathbf{x}^\star \in \mathcal{C}$ is a local optimum if there exists $R > 0$ such that

$$F(\mathbf{x}^\star) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \{\mathbf{x} \in \mathcal{C}, \|\mathbf{x} - \mathbf{x}^\star\| \leq R\}$$

- If the problem is convex, all local optimum are global.
- For non-convex function, the optimum is global only if the equation is true for all $R > 0$.



Convex Nonconvex

First order condition

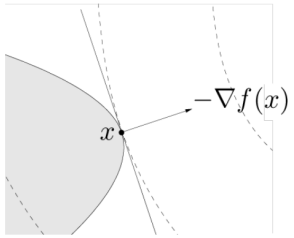
Convex and differentiable function

For the following problem

$$\min_{\mathbf{x}\in\mathcal{C}} F(\mathbf{x})$$

the feasible point $\mathbf{x}^\star \in \mathcal{C}$ is globally optimal if and only if

$$\nabla F(\mathbf{x}^\star)^\top (\mathbf{y}-\mathbf{x}) \geq 0 \quad \forall \mathbf{y} \in \mathcal{C}$$



- Any feasible direction from $\mathbf{x}^\star$ is aligned with an increasing gradient.
- If $\mathcal{C} = \mathbb{R}^d$, the condition is equivalent to

$$\nabla F(\mathbf{x}^\star) = 0$$

Empirical risk minimization

Supervised Machine learning

$$\min_f \quad \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) \tag{14}$$

- Find the function f that minimizes the average error L of prediction on a finite dataset of size N.
- Usually $f_{\boldsymbol{\theta}}$ is parametrized by $\boldsymbol{\theta} \in \mathbb{R}^n$ so the optimization is done *w.r.t.* θ.
- The objective above is called Empirical Risk Minimization, but beware of over-fitting when the model f is too complex.

Structural Risk Minimization [Vapnik, 2013]

$$\min_f \quad \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) + \lambda R(f) \tag{15}$$

- $R(f)$ is a regularization term that measure the complexity of f.
- λ is a regularization parameter that weight the regularization.

Least Square and ridge regression

Linear regression

$$\min_{\mathbf{x}} \quad \frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|^2 + \lambda\frac{1}{2}\|\mathbf{x}\|^2$$

- Objective: predict a continuous value with a linear model (regression).
- Quadratic loss : $L(y, f(\mathbf{x})) = \frac{1}{2}(y - f(\mathbf{x}))^2$
- Quadratic regularization : $R(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$.
- Smooth and strictly convex problem when $\lambda > 0$.
- Can be solved by solving a linear problem (linear equations).

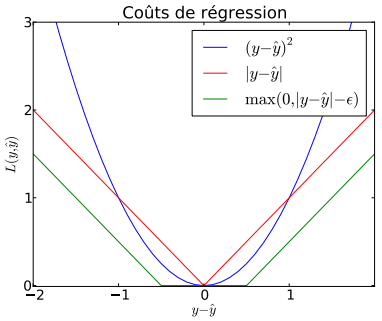
Non-linear regression

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{N}\sum_{i=1}^{N}(y_i - f_{\boldsymbol{\theta}}(\mathbf{x}_i))^2$$

- Classical formulation for regression with neural networks.
- Can be non-convex and non-smooth depending on the architecture of $f_{\boldsymbol{\theta}}$.
- Harder to regularize (what is the complexity of $f_{\boldsymbol{\theta}}$?).

Data fitting for regression

Cost	$L(y, \hat{y})$	Smooth.	Cvx.
Square	$(y-\hat{y})^2$	✓	✓
Absolute value	$\lvert y-\hat{y}\rvert$	-	✓
ϵ insensible	$\max(0, \lvert y-\hat{y}\rvert - \epsilon)$	-	✓

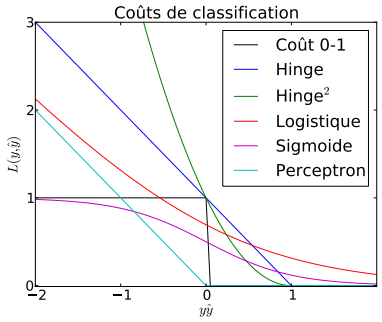


Regression problem

- **Objective**: predict a real value.
- Error if $y \neq \hat{y}$.
- **Error measure**: $|y - \hat{y}|$

Data fitting for classification

Cost	$L(y, \hat{y})$	Smooth.	Cvx.
0-1 loss	$(1 - \text{sgn}(y\hat{y}))/2$	-	-
Hinge	$\max(0, 1 - y\hat{y})$	-	✓
Squared Hinge	$\max(0, 1 - y\hat{y})^2$	✓	✓
Logistic	$\log(1 + \exp(-y\hat{y}))$	✓	✓
Sigmoid	$(1 - \tanh(y\hat{y}))/2$	✓	-
Perceptron	$\max(0, -y\hat{y})$	-	✓



Regression problem

- **Objective**: predict a binary value.
- Error when $y \neq \text{signe}(\hat{y})$ i.e. if y and $\hat{y}$ have a different sign.
- **Error measure**: $y\hat{y}$
- Non symmetric loss.

Maximum Likelihood

Maximum likelihood estimation

- p_θ is a probability distribution in $\mathbb{R}^d$.
- We have access to samples $\mathbf{x}_i$ drawn I.I.D. from the distribution.
- The likelihood for independent samples can be expressed as

$$\prod_i p_\theta(\mathbf{x}_i)$$

- The maximum likelihood estimator of θ

$$\hat{\theta} = \arg\max_\theta \prod_i p_\theta(\mathbf{x}_i)$$

- In practice one can minimize the negative log-likelihood

$$\hat{\theta} = \arg\min_\theta -\sum_i \log(p_\theta(\mathbf{x}_i))$$

That is a special case of empirical risk minimization (least square, logistic regression).

Sparsity and variable selection

Variable selection

- In supervised learning variable section aim at finding a subset $I \in \{1, \ldots, n\}$ of all variables that leads to a good prediction.
- It is a combinatorial problem *w.r.t.* the number of variables n.
- There is a compromise between number of variables and performance.

Sparsity and linear model

For a linear model the sparsity prior can be expressed as two optimization problems

$$\min_{\mathbf{x}} \; L(\mathbf{Hx}, \mathbf{y}) + \lambda \|\mathbf{x}\|_0 \qquad \text{or} \qquad \min_{\mathbf{x}, \|\mathbf{x}\|_0 \leq \tau} L(\mathbf{Hx}, \mathbf{y})$$

- $\lambda \geq 0$ and $\tau \geq 0$ are regularization parameters.
- $\|\mathbf{x}\|_0 = \sum_i 1_{|x_i|>0}$ is the number of components in $\mathbf{x}$.
- The problem can be reformulated as a Mixed Integer Program.
- Often a continuous approximation of the problem is solved (Lasso).

Lasso estimator

$$\min_{\mathbf{x}} \quad \frac{1}{2}\|\mathbf{Hx} - \mathbf{y}\|^2 + \lambda \sum_{k=1}^{d} |x_k| \tag{16}$$

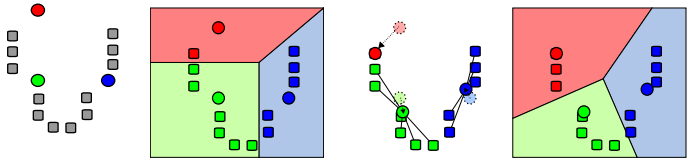
Optimization problem

- $\|\mathbf{x}\|_1 = \sum_{k=1}^d |x_k|$ is the L1 norm of vector $\mathbf{w}$.
- Objective function is non differentiable in $x_k = 0, \forall k$.
- For a large enough λ the solution of the problem is sparse.
- The problem is equivalent to

$$\min_{\mathbf{x}, \|\mathbf{x}\|_1 \leq \mu} \quad \frac{1}{2}\|\mathbf{Hx} - \mathbf{y}\|^2 \tag{17}$$

I.e. there exists a μ that leads to the same solution of the problem for a given λ.

K-means clustering

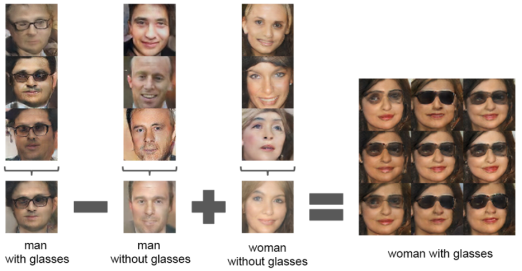


- Non convex Optimization problem:

$$\min_{\bar{\mathbf{x}}_k, \forall_k} \quad \sum_{i=1}^{N} \min_k \|\bar{\mathbf{x}}_k - \mathbf{x}_i\|^2$$

- Very simple algorithm :
  1. Update cluster membership (find closest $\bar{\mathbf{x}}_k$ for each samples)
  2. Update cluster positions $\bar{\mathbf{x}}_k$ as mean of all cluster members.
- Decrease the objective value at each iteration (can be formulated as block coordinate descent).

Generative Adversarial Networks (GAN)



Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

$$\min_G \max_D \quad E_{\mathbf{x}\sim\mu_d}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[\log(1 - D(G(\mathbf{z})))]$$

- Learn a generative model G that outputs realistic samples from data μ_d.
- Learn a classifier D to discriminate between the generated and true samples.
- Make those models compete (Nash equilibrium [Zhao et al., 2016]).
- Generator space has semantic meaning [Radford et al., 2015].

Conclusion

Machine learning and optimization

- Learning is an optimization problem.
- Design a new machine learning method $\equiv$ design a new optimization problem.
- Convexity, smoothness lead to specific solver and guarantees.

Know your optimization problems

- If convex and/or standard problems (LP,QP) $\rightarrow$ standard solvers, interior point.
- If smooth and unconstrained $\rightarrow$ Gradient descent and variants.
- If non-smooth $\rightarrow$ proximal, projected, conditional gradients.

Those are the next three parts of the course.

Bibliography I

References books for the whole course.

Convex Optimization [Boyd and Vandenberghe, 2004]

- Available freely online: `https://web.stanford.edu/~boyd/cvxbook/`.
- Perfect introduction to convex optimization (the whole book).
- Convex sets (Ch. 2), Convex functions (Ch 3), Convex problems (Ch. 4).

Elements of statistical learning [Friedman et al., 2001]

- Freely available `https://web.stanford.edu/~hastie/Papers/ESLII.pdf`
- Perfect introduction to statistical learning and machine learning.
- Most of them are optimization problems!

Nonlinear Programming [Bertsekas, 1997]

- Reference optimization book, contains also most of the course.
- Unconstrained optimization (Ch. 1), duality and lagrangian (Ch. 3, 4 ,5).

Bibliography II

Other references

Convex analysis and monotone operator theory in Hilbert spaces [Bauschke et al., 2011]

- Awesome book with lot's of algorithms, and convergence proofs.
- All definitions (convexity, lower semi continuity) in specific chapters.
- All you need to know about proximal methods.

Numerical optimization [Nocedal and Wright, 2006]

- Classic introduction to numerical optimization.
- Very detailed unconstrained optimization, specific chapters for LP and QP.

Optimization for Machine Learning [Sra et al., 2012]

- Specific chapters for precise problems (non-convex, sparsity, interior points)
- For this course: Convex with sparsity (Ch. 2), Interior points (Ch. 3).

Linear Programming [Vanderbei et al., 2015]

- Reference book of LP (Simplex, interior point)

References I

[Bauschke et al., 2011] Bauschke, H. H., Combettes, P. L., et al. (2011).
Convex analysis and monotone operator theory in Hilbert spaces, volume 408.
Springer.

[Bertsekas, 1997] Bertsekas, D. P. (1997).
Nonlinear programming.
Journal of the Operational Research Society, 48(3):334–334.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004).
Convex optimization.
Cambridge university press.

[Diamond and Boyd, 2016] Diamond, S. and Boyd, S. (2016).
Cvxpy: A python-embedded modeling language for convex optimization.
The Journal of Machine Learning Research, 17(1):2909–2913.

[Friedman et al., 2001] Friedman, J., Hastie, T., and Tibshirani, R. (2001).
The elements of statistical learning, volume 1.
Springer series in statistics New York.

References II

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680.

[Grant and Boyd, 2014] Grant, M. and Boyd, S. (2014).
CVX: Matlab software for disciplined convex programming, version 2.1.
http://cvxr.com/cvx.

[Grant et al., 2006] Grant, M., Boyd, S., and Ye, Y. (2006).
Disciplined convex programming.
Springer.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006).
Numerical optimization.
Springer Science & Business Media.

[Radford et al., 2015] Radford, A., Metz, L., and Chintala, S. (2015).
Unsupervised representation learning with deep convolutional generative adversarial networks.
arXiv preprint arXiv:1511.06434.

References III

[Sra et al., 2012] Sra, S., Nowozin, S., and Wright, S. J. (2012).
Optimization for machine learning.
Mit Press.

[Vanderbei et al., 2015] Vanderbei, R. J. et al. (2015).
Linear programming.
Springer.

[Vapnik, 2013] Vapnik, V. (2013).
The nature of statistical learning theory.
Springer science & business media.

[Zhao et al., 2016] Zhao, J., Mathieu, M., and LeCun, Y. (2016).
Energy-based generative adversarial network.
arXiv preprint arXiv:1609.03126.