Optimization & Operational Research : First Part

Ievgen Redko

January 2020 - Semester II

Topic of the course

Headline

- Mathematical background : Convex sets and derivatives.
- Convex function and their properties.
- What is a convex optimization problem ?
- Algorithms for convex optimization.

Some references

Linear Algebra

- K.B Petersen, M.S Pedersen, *The Matrix Cookbook*,2012. Available at : http ://matrixcookbook.com

Convex Optimization

- Stephen Boyd & Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2014

Mathematical Background

Norms

Given $x, y \in \mathbb{R}^n$, the inner product is given by :

$$\langle x, y\rangle = x^T y = \sum_{i=1}^{n} x_i y_i.$$

The inner product of x with itself is called the square of the norm of x

$$\langle x, x\rangle = \|x\|^2.$$

Definition

Let E be a $\mathbb{R}$-vector space, then the application $\|.\|$ is said to be a norm if for all $u, v \in E$ and $\lambda \in \mathbb{R}$

1. (positive) $\|u\| \geq 0$,
2. (definite) $\|u\| = 0 \iff u = 0$,
3. (scalability) $\|\lambda u\| = |\lambda|\|u\|$,
4. (triangle inequality) $\|u + v\| \leq \|u\| + \|v\|$.

Norms

Given $x, y \in \mathbb{R}^n$, the inner product is given by :

$$\langle x, y \rangle = x^T y = \sum_{i=1}^{n} x_i y_i.$$

The inner product of x with itself is called the square of the norm of x

$$\langle x, x \rangle = \|x\|^2.$$

Definition

Let E be a $\mathbb{R}$-vector space, then the application $\|.\|$ is said to be a norm if for all $u, v \in E$ and $\lambda \in \mathbb{R}$

1. (positive) $\|u\| \geq 0$,
2. (definite) $\|u\| = 0 \iff u = 0$,
3. (scalability) $\|\lambda u\| = |\lambda| \|u\|$,
4. (triangle inequality) $\|u + v\| \leq \|u\| + \|v\|$.

Norms

Given $x, y \in \mathbb{R}^n$, the inner product is given by :

$$\langle x, y \rangle = x^T y = \sum_{i=1}^{n} x_i y_i.$$

The inner product of x with itself is called the square of the norm of x

$$\langle x, x \rangle = \|x\|^2.$$

Definition

Let E be a $\mathbb{R}$-vector space, then the application $\|.\|$ is said to be a norm if for all $u, v \in E$ and $\lambda \in \mathbb{R}$

1. (positive) $\|u\| \geq 0$,
2. (definite) $\|u\| = 0 \iff u = 0$,
3. (scalability) $\|\lambda u\| = |\lambda| \|u\|$,
4. (triangle inequality) $\|u + v\| \leq \|u\| + \|v\|$.

Norms

Given $x, y \in \mathbb{R}^n$, the inner product is given by :

$$\langle x, y \rangle = x^T y = \sum_{i=1}^{n} x_i y_i.$$

The inner product of x with itself is called the square of the norm of x

$$\langle x, x \rangle = \|x\|^2.$$

Definition

Let E be a $\mathbb{R}$-vector space, then the application $\|.\|$ is said to be a norm if for all $u, v \in E$ and $\lambda \in \mathbb{R}$

1. (positive) $\|u\| \geq 0$,
2. (definite) $\|u\| = 0 \iff u = 0$,
3. (scalability) $\|\lambda u\| = |\lambda| \|u\|$,
4. (triangle inequality) $\|u + v\| \leq \|u\| + \|v\|$.

Norms

The norm can be seen as distance between two vectors x, y in the same vector space

$$\text{dist}(x, y) = \|x - y\|.$$

Example of usual norms :

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (Manhattan)
- $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ (Euclidean)
- $\|x\|_\infty = \max\left(|x_1|, \ldots, |x_n|\right)$
- More generally we define the norm $\|.\|_p$ for all integers p as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

Norms

The norm can be seen as distance between two vectors x, y in the same vector space

$$\text{dist}(x, y) = \|x - y\|.$$

Example of usual norms :

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (Manhattan)
- $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ (Euclidean)
- $\|x\|_\infty = \max\left(|x_1|, \ldots, |x_n|\right)$
- More generally we define the norm $\|.\|_p$ for all integers p as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

Norms

The norm can be seen as distance between two vectors x, y in the same vector space

$$\text{dist}(x, y) = \|x - y\|.$$

Example of usual norms :

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (Manhattan)
- $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ (Euclidean)
- $\|x\|_\infty = \max(|x_1|, \dots, |x_n|)$
- More generally we define the norm $\|.\|_p$ for all integers p as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

Norms

The norm can be seen as distance between two vectors x, y in the same vector space

$$\text{dist}(x, y) = \|x - y\|.$$

Example of usual norms :

- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ (Manhattan)
- $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ (Euclidean)
- $\|x\|_\infty = \max\left(|x_1|, \ldots, |x_n|\right)$
- More generally we define the norm $\|.\|_p$ for all integers p as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

Example 1/2

We will show that the Euclidean norm is a true norm. Let $x, y \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ then

1. It is obvious that $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$ is positive.
2. As $|x_i|^2 \geq 0$ then $\sum_{i=1}^n |x_i|^2 = 0$ if and only if $\forall i,\ x_i = 0$
3. Finally,

$$\begin{aligned}\|\lambda x\|_2 &= \sqrt{\sum_{i=1}^n |\lambda x_i|^2} \\ &= \sqrt{\sum_{i=1}^n |\lambda|^2 |x_i|^2} \\ &= |\lambda| \sqrt{\sum_{i=1}^n |x_i|^2}.\end{aligned}$$

Example 2/2

To prove the last point we will use the Cauchy-Schwartz Inequality :

$$\langle x, y \rangle \leq \|x\| \|y\|.$$

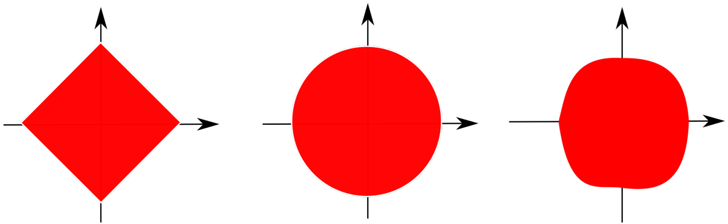
We have,

$$\begin{aligned} \|x+y\|_2^2 &= \|x\|_2^2 + 2\langle x, y \rangle + \|y\|_2^2 \\ &\leq \|x\|_2^2 + 2\|x\|_2\|y\|_2 + \|y\|_2^2 \\ &\leq \left(\|x\|_2 + \|y\|_2\right)^2. \end{aligned}$$

By taking the square root, which is an increasing function, we get the result.

Norms and Unit Ball

Unit ball for the norms $\|\|_p$ for $p = 1, 2$ and $p > 2$



Exercise

1. Represent the unit ball for the norm $\|.\|_\infty$.
2. Show that $\|x\|_1 = \sum_{i=1}^n |x_i|$ is a norm.

Correction

- The Unit Ball using the $\|.\|_\infty$ is a full square.
- We have to check the four points of the definition.

1. $\|x\|_1 = \sum_{i=1}^n |x_i| \geq 0$ by definition of the absolute value.
2. $\|x\|_1 = \sum_{i=1}^n |x_i| \geq 0 \implies x = 0$ because the sum of positive numbers is equal to zero if and only if all the terms are equal to zero.
3. $\|\lambda x\|_1 = \sum_{i=1}^n |\lambda x_i| = |\lambda| \sum_{i=1}^n |x_i| = |\lambda| \|x\|_1$.
4. $\|x + y\|_1 = \sum_{i=1}^n |x_i + y_i| \leq \sum_{i=1}^n |x_i| + \sum_{i=1}^n |y_i| = \|x\|_1 + \|y\|_1$.

Norms on matrices

It is also to define an inner product and a norms on matrices :

1. Given two matrices $X, Y \in \mathbb{R}^{m \times n}$ the **inner product** is defined by :

$$\langle X, Y \rangle = Tr\left(X^T Y\right) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} y_{ij}.$$

2. A classical norm used with matrices is the Frobenius norm :

$$\|X\|_F = \sqrt{Tr\left(X^T X\right)} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^2\right)^{1/2}.$$

What is the inner product of the symmetric matrices $X, Y \in \mathcal{S}^n(\mathbb{R})$?

Convex Sets

Definition

A set C is said to be convex if, for every $(u, v) \in C$ and for all $t \in [0, 1]$ we have :

$$tu + (1 - t)v \in C.$$

In other words, C is said to be convex if **every point on the segment connecting u and v is in the set.**

Convex Sets

Definition

A set C is said to be convex if, for every $(u, v) \in C$ and for all $t \in [0, 1]$ we have :

$$tu + (1 - t)v \in C.$$

In other words, C is said to be convex if **every point on the segment connecting u and v is in the set.**

Proposition

Let $(u_1, u_2, \ldots, u_n)$ be a set of n points belonging to a convex set C. Then for every reel numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $\sum_{i=1}^{n} \lambda_i = 1$:

$$v = \sum_{i=1}^{n} \lambda_i u_i \in C.$$

Convex Sets

Definition

A set C is said to be convex if, for every $(u, v) \in C$ and for all $t \in [0, 1]$ we have :

$$tu + (1-t)v \in C.$$

In other words, C is said to be convex if **every point on the segment connecting u and v is in the set.**

Proposition

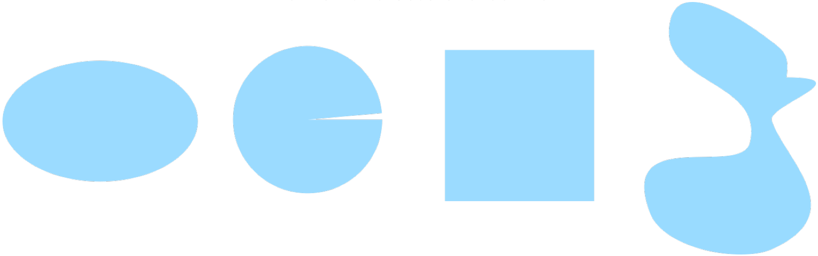
Let $(u_1, u_2, \ldots, u_n)$ be a set of n points belonging to a convex set C. Then for every reel numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $\sum_{i=1}^{n} \lambda_i = 1$:

$$v = \sum_{i=1}^{n} \lambda_i u_i \in C.$$

Every convex combination of points in a convex set is **in the convex set**.

Convex Sets

Which of the sets are convex ?



Examples of Convex Sets

1. $\mathcal{B} = \{u \in \mathbb{R}^n \mid \|u\| \leq 1\}$ is convex.
2. Every segment in $\mathbb{R}$ is convex.
3. Every hyperplane $\{x \in \mathbb{R}^n \mid a^T x = b\}$ is convex.
4. If C_1 and C_2 are two convex sets, then the intersection $C = C_1 \cap C_2$ is also convex.

Examples of Convex Sets

1. $\mathcal{B} = \{u \in \mathbb{R}^n \mid \|u\| \leq 1\}$ is convex.
2. Every segment in $\mathbb{R}$ is convex.
3. Every hyperplane $\{x \in \mathbb{R}^n \mid a^T x = b\}$ is convex.
4. If C_1 and C_2 are two convex sets, then the intersection $C = C_1 \cap C_2$ is also convex.

Exercise

1. Prove that the Euclidean Unit Ball is convex.
2. (At home) Prove that a set A is convex if and only if its intersection with any line is convex.

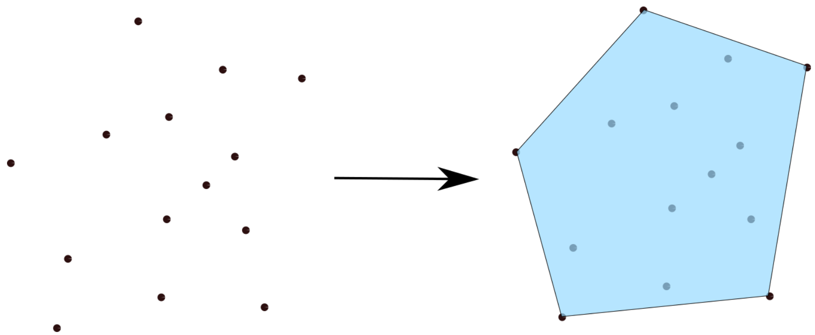
Correction

- For the first point, consider $\lambda \in [0, 1]$ and u, v two vectors in the unit ball. Then set $z = \lambda u + (1 - \lambda)v$. (i) take the norm of z, (ii) apply the triangle inequality and (iii) the scalability of the norm.
- Use the definition of convexity

Build a convex set

For a convex set and a set of point $x_1, \ldots x_n$, it is possible to build a convex set. This new set is called the convex hull $\mathcal{H}$ of a set of points

$$\mathcal{H} = \{y = \sum_{i=1}^{n} \lambda_i x_i \mid \sum_{i=1}^{n} \lambda_i = 1\}.$$



Derivative for real functions

Recall

Let $f : \mathbb{R} \to \mathbb{R}$ be continuous and $x_0 \in \mathbb{R}$. We say that f is differentiable at x_0 if the limit :

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h},$$

exists and is finite.

If f is continuously differentiable at x_0, so for $h \simeq 0$ we have

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \varepsilon(h).$$

This formula (Taylor's Formula) can be generalized to a function g n-times continuously differentiable :

$$f(x_0 + h) = f(x_0) + \sum_{i=1}^{n} \frac{h^{(i)}}{i!} f^{(n)}(x_0) + \varepsilon(h^n).$$

First order derivative

Definition

Let $f : \mathbb{R}^m \to \mathbb{R}^n$ be a C^0 application and $x \in \mathbb{R}^m$. Then f is differentiable at x_0 if it exists $J \in \mathbb{R}^{m \times n}$ such that :

$$\lim_{x \to x_0} \quad \frac{\|f(x) - f(x_0) - Jf(x_0)(x - x_0)\|}{\|x - x_0\|} = 0.$$

D is called the Jacobian of the application f.

For all i, j the elements of the matrix J are given by :

$$J_{ij}f(x_0) = \frac{\partial f_i(x)}{\partial x_j}\bigg|_{x=x_0}$$

First order derivative

Remark

Usually $f : \mathbb{R}^m \to \mathbb{R}$ so the Jacobian of the application f (also called the gradient) will be a **vector** $\nabla f(x_0)$

The gradient gives the possibility to approximate the function near the point its gradient is calculated. For all $x \in V(x_0)$ we have

$$f(x) \simeq f(x_0) + \nabla f(x_0)(x - x_0)$$

This affine approximation of the function f will help us to characterize convex functions.

First order derivative : example

Let us consider a function $f : \mathbb{R}^3 \to \mathbb{R}$ defined by

$$f(x, y, z) = 3x^2 + 2xyz + 6z + 5yz + 9xz.$$

We want to calculate the Jacobian of this function. To do so, we need to calculate : $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$. The Jacobian of f at (x, y, z) is given by :

$$J_{f(x,y,z)} = \begin{pmatrix} 6x + 2yz + 9z, & 2xz + 5z & 2xy + 6 + 5y + 9x \end{pmatrix}$$

First order derivative

Exercise

1. Let $x, y, z \in \mathbb{R}^n$. Calculate the Jacobian of the function
$$f(x,y,z) = exp(xyz) + x^2 + y + log(z).$$
2. Linear Regression. Let $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n\times d}$ and $\beta \in \mathbb{R}^d$. Calculate the derivative of the function
$$f(\beta) = \|Y - X\beta\|_2^2$$
3. Log-Sum-Exp. Let $x, b \in \mathbb{R}^n$. Calculate the derivative of the function
$$f(x) = \log \sum_{i=1}^{n} \exp\left(x_i + b_i\right)$$

Correction

• You simply have to apply the definition as it we have done in the previous example and you will have :

$$\nabla f(x,y,z) = \left(yz\exp(xyz) + 2x, xz\exp(xyz) + 1, xy\exp(xyz) + \frac{1}{z} \right).$$

• Here, you have to use the face that : $\|x\|^2 = \langle x, x \rangle$. Then you compute the derivative using the fact that f is defined as a product of two functions of β.

$$\nabla f(\beta) = -X^T(Y - X\beta) + ((Y - X\beta)^T(-X))^T = -2X^T(Y - X\beta).$$

• Remember that the Jacobian $\nabla_f = J_f$ is a vector where each entry i is equal to :

$$\nabla f(x)_i = \frac{\exp(x_i + b_i)}{\sum_{i=1}^{n} \exp(x_i + b_i)}.$$

Second order derivative

Definition

Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be a real function. Provided that this function is twice diffentiable, the second derivative H, (also called the Hessian)of f at x_0 is given by :

$$H_{ij} f(x_0) = \left.\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right|_{x=x_0},$$

and $H \in \mathbb{R}^{m \times m}$

Hessian is useful to prove that a function f is **convex** or not and also to build efficient algorithms.

Second order derivative : example

Let us consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(x, y) = 4x^2 + 6y^2 + 3xy + 2\left(\cos(x) + \sin(y)\right)$$

and calculate the Hessian of this function. We first have to calculate the Jacobian of the matrix and then the Hessian.

$$J_{f(x,y)} = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{pmatrix} = \begin{pmatrix} 8x + 3y - 2\sin(x) & 12y + 3x + 2\cos(y) \end{pmatrix}$$

$$H_{f(x,y)} = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial^2 y} \end{pmatrix} = \begin{pmatrix} 8 - 2\cos(x) & 3 \\ 3 & 12 - 2\sin(y) \end{pmatrix}$$

Second order derivative : example

Exercise

Calculate the second order derivative of the following functions :

- $f(x, y) = \log(x + y) + x^2 + 2y + 4$
- $f(x, y, z) = \dfrac{6x}{1 + y} + \exp(xy) + z$

Correction

The process is similar as in the previous example, so I only give the results.

$$H_{f(x,y)} = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial^2 y} \end{pmatrix} = \begin{pmatrix} 2 - \frac{1}{(x+y)^2} & -\frac{1}{(x+y)^2} \\ -\frac{1}{(x+y)^2} & -\frac{1}{(x+y)^2} \end{pmatrix}$$

$$H_{f(x,y)} = \left(\begin{matrix} y^2 \exp(xy) & -\frac{6}{(1+y)^2} + (xy+1)\exp(xy) \\ -\frac{6}{(1+y)^2} + (xy+1)\exp(xy) & \frac{12x}{(1+y)^3} + x^2 \exp(xy) \\ 0 & 0 \end{matrix} \right.$$

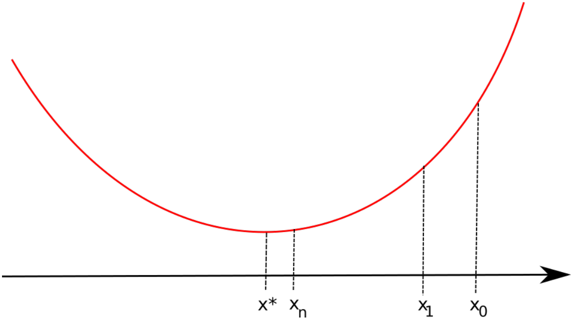
Convexity

What is a convex optimization problem ?

Given a convex function $f : \mathbb{R}^n \to \mathbb{R}$ we would solve the problem :

$$\hat{x} = \underset{x \in \mathbb{R}^n}{\arg\min} \quad f(x).$$

The aim of this part is to introduce algorithms building a series $(x_n)_{n \in \mathbb{N}}$ which converges to $\hat{x}$.



Optimization

It exists several type of optimization problem :

- convex optimization as presented before
- constraint optimization problem,
- non convex optimization problem,
- non differentiable convex optimization problem
- ...

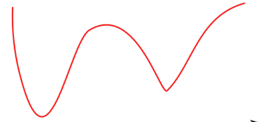
we only focus on convex optimization problem !

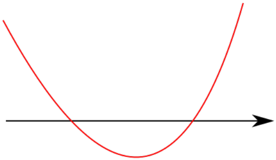
Why do we study them

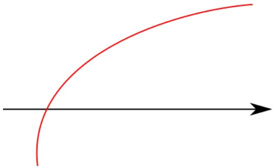
1. **Cornerstone** in modern Machine Learning.
2. Convex function admit better better algorithms than classical methods (**Gradient Descent** vs **Newton's Method**.)

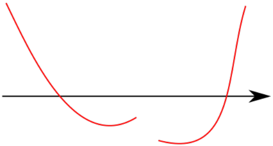
Convex Functions

Which of the following functions are convex graphically ?









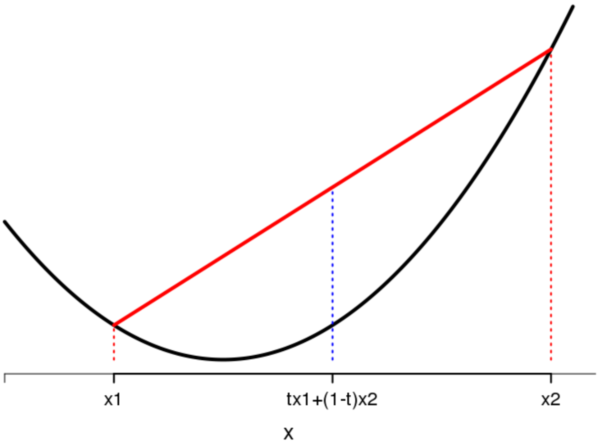
Convex Functions

Definition

Let $\mathcal{U}$ be an on empty set of a vector space ($\mathcal{U} = \mathbb{R}^n$). A function $f : \mathcal{U} \to \mathbb{R}$ is said to be convex if, for every $(u, v) \in \mathcal{U}$ and for all $t \in [0, 1]$, we have :

$$f(tu + (1-t)v) \leq tf(u) + (1-t)f(v).$$

- A linear function is convex,
- $f : \mathbb{R} \to \mathbb{R}, \quad f(x) = x^2$,
- $f : \mathbb{R} \to \mathbb{R}, \quad f(x) = \exp(x)$.



A convex function and its chord

Convex Functions and line segment

Proposition

A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex** if and only the function
$g(t) = f(x + tv)$ is convex
for all x, v such that $x + tv$ belongs to the domain of definition of f
(f is concave if and only if g is concave).

Convex Functions

Exercise

Show that the function $F : \mathbb{R} \to \mathbb{R}, \quad f(x) = x^2$ is convex.

Solution : we need to show $(tx + (1-t)y)^2 \leq tx^2 + (1-t)y^2$.

$$\iff t^2x^2 + 2t(1-t)xy + (1-t)^2y^2 \leq tx^2 + (1-t)y^2,$$

$$\iff (t^2 - t)x^2 + 2t(1-t)xy + ((1-t)^2 - (1-t))y^2 \leq 0,$$

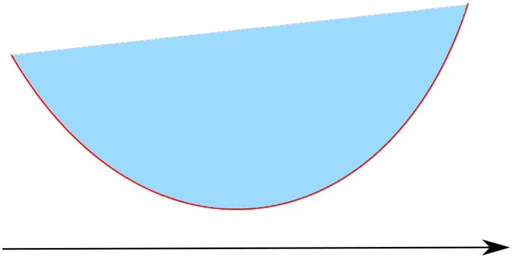
$$\iff t(t-1)x^2 - 2t(t-1)xy + t(t-1)y^2 \leq 0,$$

$$\iff t(t-1)(x-y)^2 \leq 0,$$

Convex functions

Equivalent definition

A function f is convex on $\mathcal{U}$ if and only if its epigraph E is convex, where $E = \{(x, y) \in \mathcal{U} \mid f(x) \leq y\}$.



Epigraph is the blue domain, which is convex

Concavity

Remark

Let $\mathcal{U}$ be an on empty set of a vector space ($\mathcal{U} = \mathbb{R}^n$). A function $f : \mathcal{U} \to \mathbb{R}$ is said to be concave if, for every $(u, v) \in \mathcal{U}$ and for all $t \in [0, 1]$, we have :

$$f(tu + (1-t)v) \geq tf(u) + (1-t)f(v).$$

If f is concave, then $-f$ is a convex function.

The function f defined by $f(x) = \ln(x)$ is concave.

Convex Functions

1. Given two convex functions f and g defined on $\mathcal{U}$, the sum $f+g$ is also a convex function.
2. If f is an increasing and convex function, g a convex function, then $f \circ g(x)$ is convex.
3. If f and g are convex functions, then h defined by $h(u) = \max\left(f(u), g(u)\right)$ is also convex

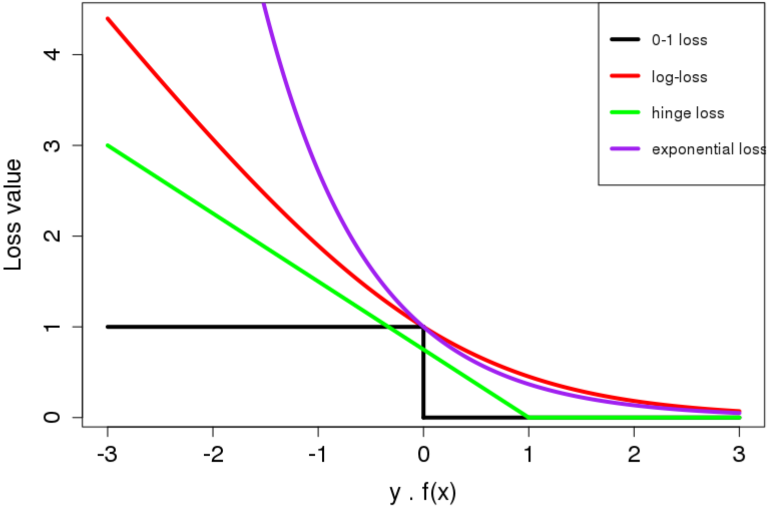
Exercise

Prove the two first points using the definition of convexity.

Correction

1. For this one, you have to notice that $(f+g)(x) = f(x) + g(x)$ and apply the definition of convexity
2. $$\begin{aligned} g(tx+(1-t)y)) &\leq tg(x)+(1-t)g(y) \\ f(g(tx+(1-t)y))) &\leq f(tg(x)+(1-t)g(y)) \\ f(g(tx+(1-t)y))) &\leq tf(g(x))+(1-t)f(g(y)) \\ f\circ g(tx+(1-t)y) &\leq tf\circ g(x)+(1-t)f\circ g(y) \end{aligned}$$

Convex Loss Functions



Convexity and differentiability

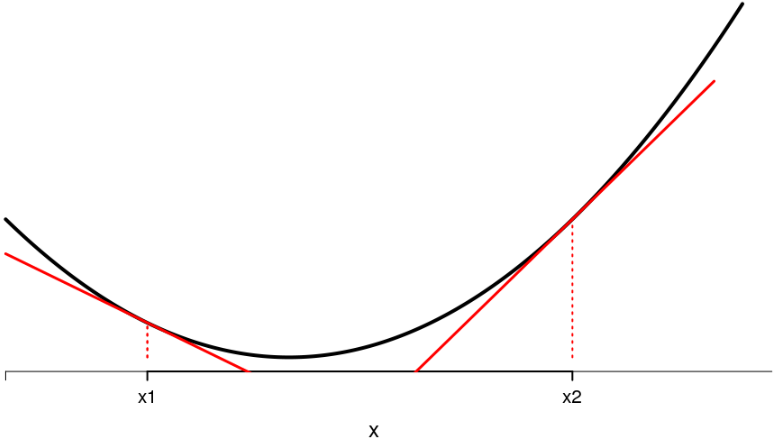
Proposition

Let f be a continuously differentiable function (C^1) on $\mathcal{U}$. Then f is convex if and only if, for all $(u, v) \in \mathcal{U}$, we have :

$$f(v) \geq f(u) + \nabla f(u)(v - u).$$

Equivalently if and only if, for all $(u, v) \in \mathcal{U}$, we have :

$$(\nabla f(v) - \nabla f(u))(v - u) \geq 0$$


x1
x2
x

Convexity and differentiability

Definition

Let f be a function of class C^2 on $\mathcal{U}$ and let H be its Hessian. Then f is **convex** if :

- $\nabla^2 f(u) \geq 0$ for all $u \in \mathcal{U}$.
- H is a positive semi definite (PSD), i.e, $\forall u \in \mathcal{U}$

$$u^T H u \geq 0.$$

Recall

A matrix H is PSD if and only if all of it's eigenvalues are **non-negative**

Convexity and differentiability

Interpretation

Positive eigenvalues imply that the **gradient is an increasing function** along each direction of the space

We consider a 2×2 matrix A :

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

where a, b, c, d are real numbers. We denote by λ_1, λ_2 the eigenvalues of this matrix (roots of the polynomial $\det(XI_2 - A)$).

Convexity and differentiability

1. We'll show why, for a 2×2 matrix, we have the following equivalence :
   A is PSD $\iff Tr(A) \geq 0$ **and** $\det(A) \geq 0$.
2. We have $\det(XI_2 - A)) = x^2 - (a+d)x + ad - bc$. The roots of this polynomial are exactly the eigenvalues of the matrix A (by definition), so

$$\det(XI_2 - A) = (x - \lambda_1)(x - \lambda_2) = x^2 - (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2.$$

   So we have, for all $x \in \mathbb{R}$:

$$x^2 - (a+d)x + ad - bc = x^2 - (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2.$$

3. It implies : $\lambda_1 + \lambda_2 = a + d = Tr(A)$ and $\lambda_1\lambda_2 = ad - bc = \det(A)$.

Convexity and differentiability

1. We'll show why, for a 2×2 matrix, we have the following equivalence :
   A is PSD $\iff Tr(A) \geq 0$ **and** $\det(A) \geq 0$.
2. We have $\det(XI_2 - A)) = x^2 - (a+d)x + ad - bc$. The roots of this polynomial are exactly the eigenvalues of the matrix A (by definition), so

$$\det(XI_2 - A) = (x - \lambda_1)(x - \lambda_2) = x^2 - (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2.$$

   So we have, for all $x \in \mathbb{R}$:

$$x^2 - (a+d)x + ad - bc = x^2 - (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2.$$

3. It implies : $\lambda_1 + \lambda_2 = a + d = Tr(A)$ and $\lambda_1\lambda_2 = ad - bc = \det(A)$.

Convexity and differentiability

1. We'll show why, for a 2×2 matrix, we have the following equivalence :
A is PSD $\iff Tr(A) \geq 0$ **and** $\det(A) \geq 0$.
2. We have $\det(XI_2 - A)) = x^2 - (a+d)x + ad - bc$. The roots of this polynomial are exactly the eigenvalues of the matrix A (by definition), so

$$\det(XI_2 - A) = (x - \lambda_1)(x - \lambda_2) = x^2 - (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2.$$

So we have, for all $x \in \mathbb{R}$:

$$x^2 - (a+d)x + ad - bc = x^2 - (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2.$$

3. It implies : $\lambda_1 + \lambda_2 = a + d = Tr(A)$ and $\lambda_1\lambda_2 = ad - bc = \det(A)$.

Convexity and differentiability

1. ($\Rightarrow$) If the eigenvalues are positive, we immediately see that both :

$$Tr(A) > 0 \quad and \quad \det(A) \geq 0.$$

2. ($\Leftarrow$) Conversely, if $\det(A) \geq 0$ it means that the two eigenvalues have the same sign. Moreover, if the trace is positive then the two eigenvalues are positive.

Convexity and differentiability

Remark

A matrix A is said to be NSD (Negative Semi-Definite) if its eigenvalues are non-positive. A 2×2 matrix A is NSD if we have :

$$Tr(A) < 0 \quad and \quad \det(A) \geq 0.$$

Examples

- If for all $i = 1, \dots, n, \quad \lambda_i \geq 0$, then $H = \text{diag}(\lambda_i)$ is PSD.
- The function $f : \mathbb{R}^n \to \mathbb{R}$ defined by $f(x_1, \dots, x_n) = \sum_{i=1}^{n} x_i^2$ is convex.

Examples

- If for all $i = 1, \ldots, n, \quad \lambda_i \geq 0$, then $H = \mathrm{diag}(\lambda_i)$ is PSD.
- The function $f : \mathbb{R}^n \to \mathbb{R}$ defined by $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i^2$ is convex.

Exercises

- Show that the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x, y) = 2x^2 + 2xy + 2y^2$ is convex.
- Show that the function $f : \mathbb{R}^3 \to \mathbb{R}$ defined by $f(x, y, z) = 5x^2 + 2\sqrt{2}xy + 6y^2 + 3z^2$ is convex.
- Show that the function $f : \mathbb{R}^n \to \mathbb{R}$ defined by $f(x) = \log\left(\sum_{i=1}^{N} e^{x_i}\right)$ is convex.

Correction 1/6

For the two first functions, you have to check that all the eigenvalues of the Hessian Matrix are non-negative. So you need : 1) to compute the Hessian Matrix of the given function and 2) to compute the eigenvalues of this last. Remember that the eigenvalues of a given matrix H are given by finding the roots of the following polynomial in λ :

$$det(H - \lambda I_d)$$

Correction 2/6

- For the first function, the Hessian Matrix is given by :

$$H_f(x,y) = \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix},$$

The eigenvalues are then given by finding the roots of the polynom :

$$det\left(H_f(x,y) - \lambda I_2\right) = det\begin{pmatrix} 4-\lambda & 2 \\ 2 & 4-\lambda \end{pmatrix} = (4-\lambda)^2 - 2^2 = (\lambda-2)(\lambda-6).$$

The eigenvalues are 2 and 6, they are non-negative so the function f is convex.

Correction 3/6

- For the second function, the Hessian Matrix is given by :

$$H_f(x,y) = \begin{pmatrix} 10 & 2\sqrt{2} & 0 \\ 2\sqrt{2} & 12 & 0 \\ 0 & 0 & 6 \end{pmatrix},$$

The eigenvalues are then given by finding the roots of the polynom :

$$det\,(H_f(x,y) - \lambda I_3) = det \begin{pmatrix} 10-\lambda & 2\sqrt{2} & 0 \\ 2\sqrt{2} & 12-\lambda & 0 \\ 0 & 0 & 6-\lambda \end{pmatrix}.$$

$$det\,(H_f(x,y) - \lambda I_3) = (6-\lambda)[(10-\lambda)(12-\lambda)-8] = (6-\lambda)(\lambda-8)(\lambda-14).$$

The eigenvalues are $6, 8$ and 14, they are non-negative so the function f is convex.

Correction 4/6

- For this last function, we will use the expression of the Jacobian previously computed :

$$J_f(x) = \frac{1}{\sum_{i=1}^{n} \exp(x_i)} (\exp(x_1, ..., \exp(x_n))$$

Then we compute the Hessian, we will seperate the diagonal terms with the non-diagonal one. For convenience, we will set $z_i = \exp(x_i)$, $Z = \sum_{i=1}^{n} \exp(x_i)$ and $z = (z_1, ..., z_n)$.

$$H_f(x, y)_{(i,j)} = \begin{cases} \dfrac{z_i Z - z_i^2}{Z^2} & if \quad i = j \\ \\ -\dfrac{z_i z_j}{Z^2} & if \quad i \neq j \end{cases}$$

Correction 5/6

Using the previous notations, we can write :

$$H_f(x,y)_{(i,j)} = \frac{1}{Z} diag(z) - \frac{1}{Z^2} zz^T.$$

To proove that this function is convex, we will show that for vector $u \in \mathbb{R}^n$ we have $u^T H_f u \geq 0$.

$$u^T H_f u = \frac{1}{Z^2} \left(\left(\sum_{i=1}^{n} u_i^2 z_i \right) \left(\sum_{i=1}^{n} z_i \right) - \left(\sum_{i=1}^{n} u_i z_i \right)^2 \right).$$

We need to show that is expression is non-negative. For that, we use the **Cauchy-Schwarz Inequality**. So we will introduce inner product and norms.

Correction 6/6

Note that : $\sum_{i=1}^n u_i^2 z_i = \|u_i\sqrt{z_i}\|_2^2$, $\sum_{i=1}^n z_i = \|\sqrt{z_i}\|_2^2$ and $\left(\sum_{i=1}^n u_i z_i\right)^2 = \|u_i z_i\|_2^2$. So that :

$$u^T H_f u = \frac{1}{Z^2}\left(\|u\sqrt{z}\|\|\sqrt{z}\| - \langle u\sqrt{z}, \sqrt{z}\rangle^2\right).$$

We can bound the inner product as follow :

$$\langle u\sqrt{z}, \sqrt{z}\rangle^2 \leq \|u\sqrt{z}\|\|\sqrt{z}\|.$$

We conclude that :

$$u^T H_f u \geq 0.$$

Convex Optimization

Condition of Optimality

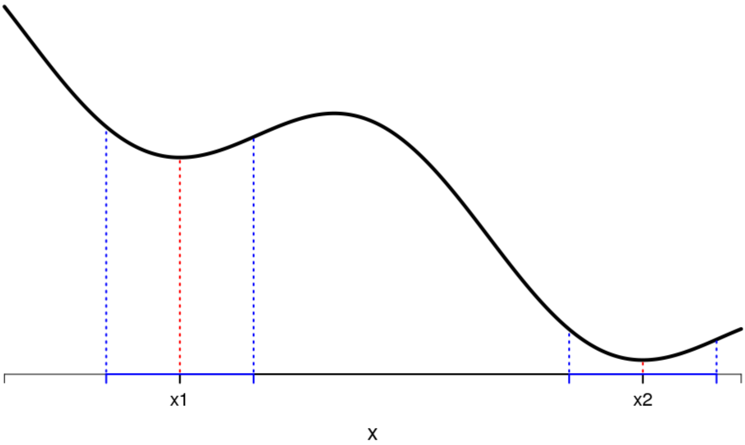
Definition

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function. We say that $u \in \mathbb{R}^n$ is a local minimum of f if it exists a neighborhood $V \subset \mathbb{R}^n$ of u such that :

$$f(u) \leq f(v), \quad \forall v \in V.$$

u is a global minimum of the function f if and only if :

$$f(u) \leq f(v), \quad \forall v \in \mathbb{R}^n.$$



- x_1 and x_2 are two local minima of f.
- x_2 is the global minimum of the function f

Condition of Optimality

Proposition : - Euler's Equation -

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and differentiable at $u \in \mathbb{R}^n$. If u is a local minimum then we have : $\nabla f(u) = 0$.

Condition of Optimality

Proposition : - Euler's Equation -

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and differentiable at $u \in \mathbb{R}^n$. If u is a local minimum then we have : $\nabla f(u) = 0$.

Proof : In fact, using the definition : $\forall v \in \mathbb{R}^n, \exists t > 0$ such that $u + tv \in V$ a neighborhood of u.

$$\begin{array}{rcl} & f(u) & \leq f(u+tv) = f(u) + \nabla f(u)(tv) + tv\,\varepsilon(tv), \quad t \ll 1 \\ \iff & 0 & \leq \nabla f(u)(tv) + tv\,\varepsilon(tv) \end{array}$$

Dividing by $t > 0$ and taking the limit $t \to 0$ we have : $0 \leq \nabla f(u)v$.
Same thing by replacing $v \to -v$ we have $0 \leq -\nabla f(u)v$.
So $\forall v \in \mathbb{R}^n, \quad \nabla f(u)v = 0 \Rightarrow \nabla f(u) = 0$.

Condition of Optimality

The solution of *Euler's Equation* gives us the points where the function f reaches a local extremum (a minimum or maximum (local or global)).

Given a solution u of $\nabla f(u) = 0$, we can say that :

- u is local minimum if $\nabla^2 f(u) = H_f(u) \geq 0$, i.e. the Hessian matrix evaluated at the point u is PSD. This point is a global minimum if the function is convex everywhere or if for all $v \neq u$ we have $f(u) \leq f(v)$.
- u is local maximum if $\nabla^2 f(u) = H_f(u) \leq 0$, i.e. the Hessian matrix evaluated at the point u is NSD. This point is a global maximum if the function is concave everywhere or if for all $v \neq u$ we have $f(u) \geq f(v)$.

Condition of Optimality

Definition

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\mathcal{U}$ a non empty set. We say that f has a relative minimum u if

$$f(u) \leq f(v), \quad \forall v \in \mathcal{U}.$$

Proposition : - Euler's Inequality -

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\mathcal{U}$ a non empty and convex set. Furthermore, let $u \in \mathcal{U}$ be a relative minimum of f. If f is differentiable at u we have : $\nabla f(u)(v - u) \geq 0$ $\forall v \in \mathcal{U}$.

Exercise

- Let f defined by $f(x, y) = (4 - 2y)^2 + 5x^2 + x + 3y + 4xy$
  1. Is the function f convex ?
  2. What is the global minimum of f ?
- Let f defined by $f(x, y) = 2x^2 + 4(y - 2)^2 + 4x + 6y - 2xy + 2y^3$.
  1. Is f convex ?
  2. Give a condition on y so that f is convex.
  3. (Optional) For the previous condition on y, find the local minimum of f

1. The function f is convex. In fact, we have :

$$H_{f(x,y)} = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial^2 y} \end{pmatrix} = \begin{pmatrix} 10 & 4 \\ 4 & 8 \end{pmatrix}.$$

Because f is convex, if we find (x, y) such that $\nabla f(x, y) = 0$ then (x, y) is the Argmin of f.

$$J_{f(x,y)} = \begin{pmatrix} 10x + 4y + 1, & 4x + 8y - 13 \end{pmatrix} = (0, 0).$$

The solution is $(x, y) = (-\frac{15}{16}, \frac{67}{32})$.

2) Same as before, we calculate the Hessian matrix :

$$H_{f(x,y)} = \begin{pmatrix} 4 & -2 \\ -2 & 12y+8 \end{pmatrix}.$$

We have $Tr(H) = 12y + 12$ and $det(H) = 48y + 28$. These quantities are both positie if and only if $y \geq -\frac{28}{48} = -\frac{7}{12}$.

So the function is not convex on $\mathbb{R}^2$, but it is on $\mathbb{R} \times [-\frac{7}{12}, \infty[$.

- You have to solve the following system :

$$\begin{aligned} 4x + 4 - 2y &= 0, \\ 6y^2 + 8y - 2x - 10 &= 0. \end{aligned}$$

$$\begin{aligned} 4x + 4 - 2y &= 0, \\ 6y^2 + 7y - 8 &= 0. \end{aligned}$$

You solve the following system, keeping the appropriate value of y and then you calculate x.

Convex Problems

The basic formulation

Given a vector space E and a function $f : E \to \mathbb{R}$, an optimization problem consists of solving the following problem :

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

• The function f is sometimes called the cost function (ie, cost for a company to store goods).

• Most of times, we want to minimize the function f under some constraints.

Linear Regression 1/3

Let us first consider the linear regression :

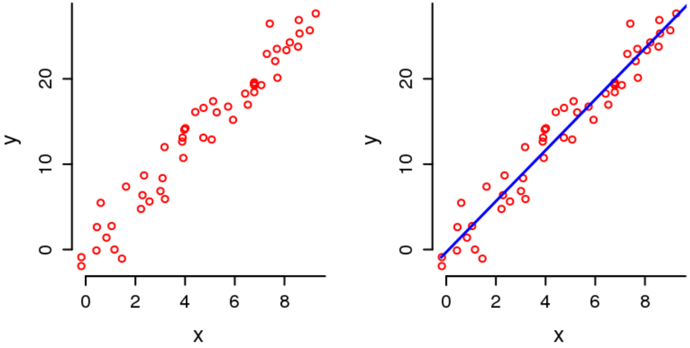
• Given a response vector $Y \in \mathbb{R}^n$ and feature vectors $X = (x_1, \ldots, x_n)^T, x_i \in \mathcal{R}^m$ where $m + 1 < n$.
We'd like to find a vector β that explain the value of Y using X with the following model

$$Y = X\beta + \varepsilon, \quad where \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

• ε represent the error due to the model. To find the best vector β we have to minimize this error, i.e. to solve :

$$\underset{\beta \in \mathbb{R}^{m+1}}{min} \varepsilon \|Y - X\beta\|^2$$

Linear Regression 2/3



Linear Regression 3/3

We easily check that is problem is convex :

$$\nabla_\beta \, \varepsilon = -2X^T(Y - X\beta),$$

and

$$\nabla_\beta^2 = 2X^T X,$$

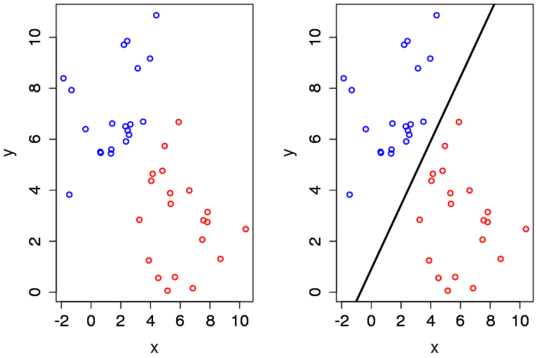
which is positive semi definite.
The solution given by

$$\beta = (X^T X)^{-1} X^T Y.$$

Analytic solution exists but this is not always the case

Logistic regression 1/2

We want to find a model that predict the class of our data.



Logistic Regression 2/2

- To predict the class of the individual we use a model of the form :

$$g(x,a) = \log\left(\frac{\mathbb{P}(X \mid Y=1)}{1-\mathbb{P}(X \mid Y=1)}\right) = a_0 + a_1x_1 + \ldots + a_mx_m.$$

- Solved by maximizing the (log-)likelihood of our data :

$$l(x,a) = \sum_{i=1}^{n} y_i \log(p_i) + (1-y_i)\log(1-p_i),\ p_i = \frac{1}{1+\exp(-\sum_{j=1}^{m} a_j x_{ij})}.$$

No analytic solution, we need a way to **approximate it** step by step.

Algorithms

Setup

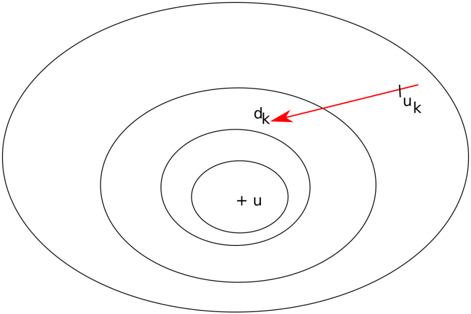
Given a function f and a non empty set $\mathcal{U}$ and **knowing there is a solution** to the problem : $f(u) = \min_{v \in \mathcal{U}} f(v)$.

Idea : build a series $(u_k)_{k\in\mathbb{N}}$ which converges to u.

Setup

Given a function f and a non empty set $\mathcal{U}$ and **knowing there is a solution** to the problem : $f(u) = \min_{v \in \mathcal{U}} f(v)$.

Idea : build a series $(u_k)_{k \in \mathbb{N}}$ which converges to u.

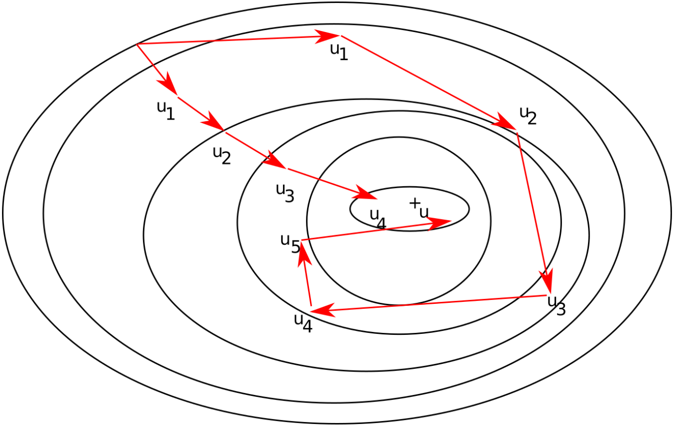


Algorithm :

- Take an initial value u_0.
- $u_k \to u_{k+1}$: Choose a direction d_k and minimize the function f along this direction.
- Solve $\arg\min_{\rho>0} f(u_k - \rho d_k) = \rho_k$
- $u_{k+1} = u_k - \rho_k d_k$

Direction of descent

How to choose the direction d_k ?



Some ways seem to be **faster** than others to reach the solution

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Direction of descent

1. Recall that

$$f(u_k - \rho d_k) = f(u_k) - \rho\langle \nabla f(u_k), d_k\rangle + \rho\varepsilon(\rho)$$

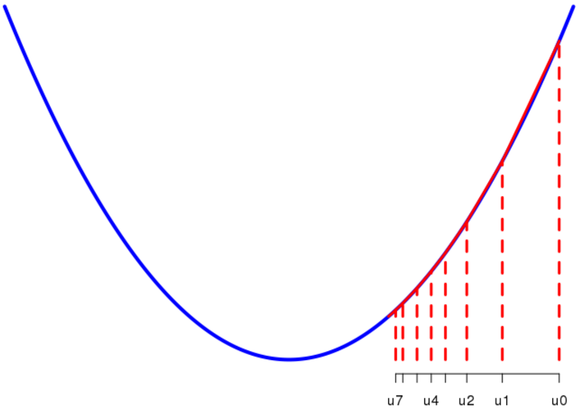
   when ρ is close to 0
2. To minimize f we choose d_k that maximizes $\langle \nabla f(u_k), d_k\rangle$
3. Due to Cauchy-Scwhartz Inequality, we have $d_k = \nabla f(u_k)$ (assuming $\|d_k\| = 1$)
4. Leads to the algorithm
   - Choose u_0 to initialize the algorithm,
   - set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$ for $\rho_k > 0$
   - till $\|\nabla f(u_k)\| \leq \varepsilon$.

Summing up

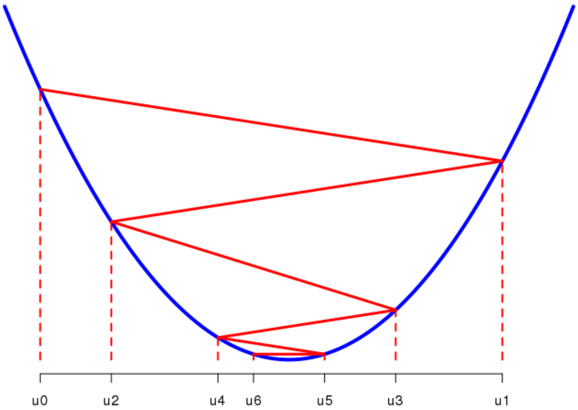
1. There exists several ways to use the gradient
2. We focus on gradient descent algorithms and their variants.
   - **Gradient Descent, Line Search, Newton's Method,...**

Other algorithms that **do not rely on** the derivatives of the function.

Gradient descent : choose the step 1/3



Gradient descent : choose the step 2/3



Gradient descent : choose the step 3/3

• If the step is too large, the sequence oscillates near the optimum.

• If the step is too small, the algorithm needs a large number of iterations.

Can **choose the step** for the gradient descents method **optimally** !

Gradient descent : with optimal step

Idea : choose the step that minimizes the objective function along a given direction.

Gradient descent : with optimal step

Idea : choose the step that minimizes the objective function along a given direction.

- Choose u_0 to initialize the algorithm,
- for $k = 0, 1, \dots$ solve $\arg\min_{\rho>0} f(u_k - \rho\nabla f(u_k))$,
- set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$
- till $\|\nabla f(u_k)\| \leq \varepsilon$.

Gradient descent : with optimal step

Idea : choose the step that minimizes the objective function along a given direction.

- Choose u_0 to initialize the algorithm,
- for $k = 0, 1, \ldots$ solve $\arg\min_{\rho > 0} f(u_k - \rho \nabla f(u_k))$,
- set $u_{k+1} = u_k - \rho_k \nabla f(u_k)$
- till $\|\nabla f(u_k)\| \leq \varepsilon$.

This algorithm is called the Gradient Descent with optimal step.

Gradient descent : with optimal step

Definition

Let f be a convex and continuously differentiable function on $\mathbb{R}^n$. We say that f is strongly convex or α-elliptical if it exists $\alpha > 0$ such that

$$\langle \nabla f(v) - \nabla f(u), v - u \rangle \geq \alpha \|v - u\|, \ \forall u, v \in \mathbb{R}^n$$

What can we say about $\langle \nabla f(u_{k+1}), \nabla f(u_k) \rangle$ based on $\rho_k = \arg\min_{\rho > 0} f(u_k - \rho d_k)$?

Gradient descent : with optimal step

If ρ_k minimize $f(u_k - \rho_k d_k)$ we have :

$$\begin{aligned} & \frac{\partial}{\partial \rho} f(u_k - \rho \nabla f(u_k))\big|_{\rho=\rho_k} = 0, \\ \iff & \langle \nabla f(u_k - \rho_k \nabla f(u_k)), \nabla f(u_k) \rangle = 0, \\ \iff & \langle \nabla f(u_{k+1}), \nabla f(u_k) \rangle = 0. \end{aligned}$$

The last equality is called the optimality condition.

Proposition

If f is a **strongly convex** then GD with optimal step converges

Gradient descent : with optimal step

Let A be a symmetric and PSD and $b \in \mathbb{R}^n$. We want to optimize

$$f(v) = \frac{1}{2}\langle Av, v\rangle - \langle b, v\rangle$$

- Calculate the gradient : $\nabla f(u_k) = Au_k - b$
- We then have to solve : $\rho_k = \arg\min_{\rho>0} f(u_k - \rho d_k)$. The optimality condition gives us : $\langle \nabla f(u_k), \nabla f(u_{k+1})\rangle = 0$

$$\begin{aligned}
\nabla f(u_{k+1}) &= Au_{k+1} - b \\
&= A(u_k - \rho_k(Au_k - b) - b \\
&= Au_k - b - \rho_k A(Au_k - b)
\end{aligned}$$

Gradient descent : with optimal step

$$
\begin{aligned}
\Rightarrow \langle Au_k - b, Au_k - b - \rho_k A(Au_k - b)\rangle &= 0\\
\Rightarrow \langle Au_k - b, Au_k - b\rangle &= \langle Au_k - b, \rho_k A(Au_k - b)\rangle\\
\Rightarrow \rho_k &= \frac{\langle Au_k - b, Au_k - b\rangle}{\langle Au_k - b, A(Au_k - b)\rangle}
\end{aligned}
$$

We finally have the following algorithm :

- Initialize $u_0 \in \mathbb{R}^n$
- At each step, calculate $\rho_k = \dfrac{\|Au_k - b\|^2}{\|Au_k - b\|_A^2}$
- Set $u_{k+1} = u_k - \rho_k(Au_k - b)$
- Stop if $\|\nabla J(u_{k+1})\| = \|Au_{k+1} - b\| \leq \epsilon$

Gradient descent : with optimal step

Exercise

Consider the matrices $A = \begin{pmatrix} 6 & 2 \\ 2 & 4 \end{pmatrix}$ and $b = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ and the application f defined by $f(v) = \langle Av, v \rangle + \langle b, v \rangle$

1. Explain why f in convex.
2. Solve the problem $u = \underset{v \in \mathbb{R}^2}{\arg\min}\ f(v)$.
3. For a given vector u_k, calculate ∇f_{u_k} and ρ_k.
4. Implement the presented method to solve this problem.

Correction

- f is defined as a quadratic function where A is PSD, so f is convex.
- We have to solve :

$$J_{f(x,y)} = \begin{pmatrix} 12x + 4y + 2, & 4x + 8y + 3 \end{pmatrix} = (0, 0).$$

The solution is $\left(-\frac{1}{20}, -\frac{7}{20}\right)$.

- Let set $u_k = (v_1, v_2)$ then :

$$\nabla f_{u_k} = \begin{pmatrix} 12v_1 + 4v_2 + 2, & 4v_1 + 8v_2 + 3 \end{pmatrix},$$

and $\rho_k = \dfrac{\|2Au_k - b\|_2^2}{\|2Au_k - b\|_A^2}$

Exercise

Let f be the function defined by : $f(x,y) = 4x^2 - 4xy + 2y^2$.

1. Is the function f convex ?
2. Apply the gradient descent with optimal step to calculate the three first steps of the algorithm using $(x_0, y_0) = (1, 1)$.

Correction 1/3

- The function f can be rewritten as : $f(u) = \frac{1}{2}u^T Au - b^T u$, where $b = (0,0)^T$ and $A = \begin{pmatrix} 8 & -4 \\ -4 & 4 \end{pmatrix}$. The function f is a quadratic function, furthermore the matrix A is PSD so the function f is convex.
- The optimal learning rate is given by :

$$\rho_k \frac{\|Au_k - b\|_2^2}{\|Au_k - b\|_A^2},$$

where the matrix A and the vector b were previously introduced.

Correction 2/3

- The function f can be rewritten as : $f(u) = \frac{1}{2}u^T A u - b^T u$, where $b = (0\ 0)^T$ and $A = \begin{pmatrix} 8 & -4 \\ -4 & 4 \end{pmatrix}$. The function f is a quadratic function, furthermore the matrix A is PSD so the function f is convex.
- The optimal learning rate is given by :

$$\rho_k = \frac{\|Au_k - b\|_2^2}{\|Au_k - b\|_A^2},$$

where the matrix A and the vector b were previously introduced. Recall that the process is defined by :

$$u_{k+1} = u_k - \rho_k \nabla f(u_k).$$

We will now apply this process to compute the three first iterations.

Correction 3/3

1. For the first iteration : $\rho_0 = \frac{\|Au_0\|_2^2}{\|Au_0\|_A^2} = \frac{16}{128} = \frac{1}{8}$. And $\nabla f(u_0) = Au_0 = (4\ 0)^T$.
$$u_1 = (1\ 1)^T - \frac{1}{8}(4\ 0)^T = (0.5\ 1)^T.$$
2. For the second iteration : $\nabla f(u_1) = Au_1 = (0\ 2)^T$. The learning rate is given by : $\rho_1 = \frac{\|Au_1\|_2^2}{\|Au_1\|_A^2} = \frac{4}{16} = \frac{1}{4}$. Thus u_2 is given by :
$$u_2 = (0.5\ 1)^T - \frac{1}{4}(0\ 2)^T = (0.5\ 0.5)^T.$$
3. For the third iteration : $\nabla f(u_2) = Au_2 = (2\ 0)^T$. The learning rate is given by : $\rho_2 = \frac{\|Au_2\|_2^2}{\|Au_2\|_A^2} = \frac{4}{32} = \frac{1}{8}$. Thus u_3 is given by :
$$u_3 = (0.5\ 0.5)^T - \frac{1}{8}(2\ 0)^T = (0.25\ 0.5)^T.$$

Gradient Descent : Armijo Criterium

Idea : use a linear search to find the **learning rate**.
Given a $\theta \in]0,1[$, choose the greatest ρ such that :

$$f(u_k - \rho \nabla f(u_k)) \leq f(u_k) - \theta\rho \|\nabla f(u_k)\|^2.$$

At each step, we reduce the function's value of at least $\theta\|\nabla f(u_k)\|$.

Gradient Descent : Armijo Criterium

Idea : use a linear search to find the **learning rate**.
Given a $\theta \in]0,1[$, choose the greatest ρ such that :

$$f(u_k - \rho \nabla f(u_k)) \leq f(u_k) - \theta \rho \|\nabla f(u_k)\|^2.$$

At each step, we reduce the function's value of at least $\theta \|\nabla f(u_k)\|$.

Armijo's condition :

- Choose $\alpha_0 > 0$ and $0 < \theta < 1$,
- Choose the greatest $s \in \mathbb{Z}$ such that :

$$f(u_k - \alpha_0 2^s \nabla f(u_k)) \leq f(u_k) - 2^s \alpha_0 \theta \|\nabla f(u_k)\|^2.$$

- Set $u_{k+1} \leftarrow u_k - \alpha_0 2^s \nabla f(u_k)$.

Gradient Descent : Armijo Criterium and Wolfe's Criteria

Theorem

If the function f is **strictly convex** and if its gradient ∇f is **Lipschitz**, then the Armijo's algorithm **converge**.

If we add the following condition to the previous one, given $0 < \theta < \eta < 1$:

$$\langle \nabla f(u_k), \nabla f(u_k - \rho \nabla f(u_k)) \rangle \geq \eta \|\nabla f(u_k)\|^2,$$

we get the Wolfe's Criteria

Conjugate Gradient

Definition

Let A be a **symmetric PD** matrix and u, v two vectors. u, v are **conjugate** with respect to A if

$$\langle Au, v \rangle = 0$$

Conjugate Gradient

Definition

Let A be a **symmetric PD** matrix and u, v two vectors. u, v are **conjugate** with respect to A if

$$\langle Au, v \rangle = 0$$

Let A be a **symmetric PD** matrix and f the function defined by

$$f(v) = \frac{1}{2}\langle Av, v \rangle - \langle b, v \rangle.$$

The objective is to build a series of **conjugate descent direction**

Conjugate Gradient

• Let $u_0 \in \mathbb{R}^n$, define a first direction of descent $d_0 = \nabla f(u_0)$ and minimize f along this direction :

$$\arg\min_{\alpha_0} f(u_0 - \alpha_0 d_0).$$

• Solving this problem we get :

$$\alpha_0 = \frac{\langle \nabla f(u_0), d_0 \rangle}{\langle A d_0, d_0 \rangle}.$$

• We set $u_1 = u_0 - \alpha_0 d_0$
• To build $d_1 = \nabla f(u_1) + \beta_0 d_0$, we need to find the value of $\beta_0 \in \mathbb{R}$ such that

$$\langle A d_1, d_0 \rangle = 0.$$

Conjugate Gradient

- We then have to solve $\langle A\nabla f(u_1), d_0\rangle + \langle A\beta_0 d_0, d_0\rangle = 0$. The solution is given by

$$\beta_0 = -\frac{\langle A\nabla f(u_1), d_0\rangle}{\langle Ad_0, d_0\rangle}.$$

Once it's done, you'll do as before.

You set $\alpha_1 = \arg\min_{\alpha} f(u_1 - \alpha d_1)$.

Set $u_2 = u_1 - \alpha_1 d_1$. And so on ...

Conjugate Gradient : Summary

Algorithm :

- Choose $u_0 \in \mathbb{R}^n$ and $d_0 = \nabla f(u_0)$.
- Set $\alpha_0 = \dfrac{\langle \nabla f(u_0), d_0 \rangle}{\langle Ad_0, d_0, \rangle}$ and $u_1 = u_0 - \alpha_0 d_0$.
- $\beta_0 = -\dfrac{\langle A\nabla f(u_1), d_0 \rangle}{\langle Ad_0, d_0 \rangle}$.

For $k \geq 1$ do,

- Set $d_k = \nabla f(u_k) + \beta_{k-1} d_{k-1}$.
- Set $\alpha_k = \dfrac{\langle \nabla f(u_k), d_k \rangle}{\langle Ad_k, d_k, \rangle}$ and $u_{k+1} = u_k - \alpha_k d_k$.
- Set $\beta_k = \dfrac{\langle A\nabla f(u_{k+1}), d_k \rangle}{\langle Ad_k, d_k \rangle}$

Untill $\|\nabla f(u_{k+1})\| \leq \varepsilon$.

Conjugate Gradient : Results

Proposition

For all $1 \leq k \leq n$ such that $\nabla f(u_0), \ldots, \nabla f(u_n)$ are non equal to zero, we have the following relations for all $0 \leq l \leq k-1$:

$$\langle \nabla f(u_k), \nabla f(u_l) \rangle = 0$$

and

$$\langle A d_k, d_l \rangle = 0.$$

Theorem

If A is a symetric positive and definite matrix, then the conjugate gradient method converges with **at most** n **steps**.

Try to prove the proposition by induction at home

Newton's Method

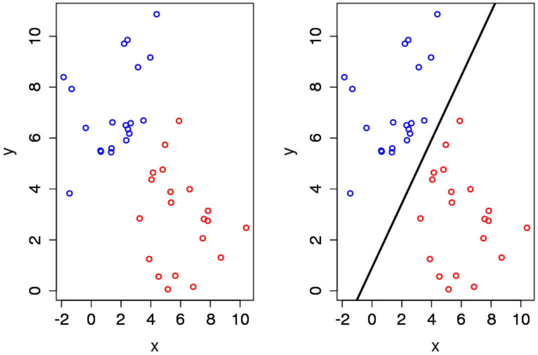
The Newton's Method is a gradient descent algorithm that refines the direction of the descent as follows :

$$u_{k+1} \leftarrow u_k - \left(\nabla^2 f(u_k)\right)^{-1} \cdot \nabla f(u_k).$$

- ✓ Requires less iterations to converge
- × Requires the inverse of the Hessian of the function we want to optimize ($\Theta(n^3)$).
- × The Hessian is not always invertible at a given point.

Newton's Method

Let's come back to the logistic regression.
We want to find a model that predict the class of our data.



→ An example of straight line that separate the two classes using logistic regression.

Newton's Method

For Logistic Regression, we want to maximize $l(x, a)$ with a **possible** solution given by solving the equation :

$$\nabla_a l(x, a) = \nabla_a \left(\sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right) = 0,$$

where $p = \left(1 + \exp(-a^T x)\right)^{-1}$.

Explain why the log-likelihood is **concave**. Calculate the **first and second derivatives** of the function l.

Newton's Method

If we apply the Newton's Method to the logistic regression we have

$$\nabla_a l(x,a) = \sum_{i=1}^{n} (y_i - p_i)\, x_i, \quad \nabla_a^2 l(x,a) = -\sum_{i=1}^{n} p_i(1-p_i) x_i x_i^T$$

We can then write the algorithm :

- Choose a_0,
- Calculate $\nabla_a l(x,a)$ and $\left(\nabla_a^2 l(x,a)\right)^{-1}$
- Set $a_{k+1} \leftarrow a_k - \left(\nabla_a^2 l(x,a)\right)^{-1} \nabla_a l(x,a)$
- Stop when $\|\nabla_a l(x,a)\| \leq \varepsilon$.

Quasi-Newton's Method : Motivation

Idea : avoid calculating the inverse of the Hessian matrix H_k^{-1} as follows :

$$\begin{aligned} u_{k+1} &= u_k - M_k \nabla f(u_k), \\ M_{k+1} &= M_k + C_k. \end{aligned}$$

Approximate the H_k^{-1} by matrix M_k at which, we add a matrix of correction C_k at each step

Quasi-Newton's Method : Motivation

Recall that :

$\nabla f(u_k) = \nabla f(u_{k+1}+(u_k-u_{k+1})) \sim \nabla f(u_{k+1})+\nabla^2 f(u_{k+1})(u_k-u_{k+1})$,

we then have :

$$\left(\nabla^2 f(u_{k+1})\right)^{-1} \left(\nabla f(u_{k+1}) - \nabla f(u_k)\right) \sim u_{k+1} - u_k.$$

If we set :

$$M_{k+1} = \left(\nabla^2 f(u_{k+1})\right)^{-1},\ \gamma_k = \nabla f(u_{k+1}) - \nabla f(u_k)$$

and $\delta_k = u_{k+1} - u_k$, we get the Quasi Newton's Condition :

$$M_{k+1}\gamma_k = \delta_k$$

Quasi-Newton's Method : Davidon-Fletcher-Powell

- Assume C_k is of rank 1, ie, C_k as $v_k v_k^T$ where $v_k \in \mathbb{R}^n$.
- The update becomes :

$$M_{k+1} = M_k + v_k v_k^T$$

.

- The Quasi Newton's Condition gives :

$$\begin{aligned} (M_k + v_k v_k^T)\gamma_k &= \delta_k, \\ M_k\gamma_k + v_k v_k^T\gamma_k &= \delta_k, \\ v_k v_k^T\gamma_k &= \delta_k - M_k\gamma_k, \\ v_k &= \frac{\delta_k - M_k\gamma_k}{v_k^T\gamma_k}. \end{aligned}$$

And the second line gives us :

$$v_k^T\gamma_k = \left(\gamma_k\delta_k - \gamma_k M_k\gamma_k\right)^{1/2}.$$

Quasi-Newton's Method : Broyden Algorithm

Broyden Algorithm

Algorithm

- Initialize $u_0 \in \mathbb{R}^n$ and M_0 (usually $M_0 = Id$),
- for $k \geq 0$ do
  - set $\rho_k = \arg\min_{\rho \in \mathbb{R}} f(u_k - \rho M_k \nabla f(u_k))$,
  - set $u_{k+1} = u_k - \rho_k M_k \nabla f(u_k)$,
  - set$M_{k+1} = M_k + \dfrac{(\delta_k - M_k\gamma_k)(\delta_k - M_k\gamma_k)^T}{(\delta_k - M_k\gamma_k)^T\gamma_k}$,

Untill $\|\nabla f(u_{k+1})\| \leq \varepsilon$.

Quasi-Newton's Method : Broyden-Fletcher-Goldfarb-Shanno

- Assume C_k is of rank 1, ie, C_k as $v_k v_k^T$ where $v_k \in \mathbb{R}^n$.
- The inverse of the Hessian, at each step, is then approximated by :

$$M_{k+1} = M_k + \left[1 + \frac{\langle M_k\gamma_k, \gamma_k\rangle}{\langle \delta_k, \gamma_k\rangle}\right] \frac{\delta_k \delta_k^T}{\langle \delta_k, \gamma_k\rangle} - \frac{\langle \delta_k, \gamma_k\rangle M_k + M_k\gamma_k\delta_k^T}{\langle \delta_k, \gamma_k\rangle}.$$

The algorithm is the same as the previous one.

Conclusion

- **Gradient descent with a constant learning rate :**
  - ✓ Easy to implement
  - × Convergence depends on the value of the learning rate
- **Gradient descent with an optimal step :**
  - ✓ Faster then simple gradient descent
  - × More costly in terms of time
- **Newton's Method :**
  - ✓ Faster than the two others.
  - ✓ Requires less iterations.
  - × Requires to invert the Hessian matrix

To go further

1. A more advanced **Adam algorithm** (used currently for DNNs)
2. **Projected gradient descent** seen later in the course