

Homework 1

Convex Optimization

Matthieu Boyer

27 décembre 2024

1 Exercises

Convex Sets

- Let $\mathcal{C} = \{x \in \mathbb{R}_+^2 \mid x_1 x_2 \geq 1\}$. We can see that $\mathcal{C} = \{(x_1, x_2) \in \mathbb{R}_{+*}^2 \mid x_1 \geq \frac{1}{x_2}\}$. Thus, \mathcal{C} is the epigraph of $x \mapsto \frac{1}{x}$ which is convex since its second derivative is $x \mapsto \frac{2x}{x^4}$. Thus, \mathcal{C} is convex.
- In general, this set is not convex. Indeed, for sets in \mathbb{R} , take $S = \{-1, 1\}, T = \{0\}$. Then the set of points closer to S than to T is $\{x \in \mathbb{R} \mid x \leq -0.5 \vee x \geq 0.5\}$ which is not an interval and thus not convex.
- Let us take $x_1, x_2 \in \{x \mid x + S_2 \subseteq S_1\}$. For $x = \lambda x_1 + (1 - \lambda) x_2$ consider $x + y$ for $y \in S_2$:

$$x + y = \lambda x_1 + (1 - \lambda) x_2 + y = \lambda(x_1 + y) + (1 - \lambda)(x_2 + y)$$

Since $x_i + S_2 \subseteq S_1$ for $i = 1, 2$, and since S_1 is convex, for all $y \in S_2$ the above sum is in S_1 and thus $x + S_2 \subseteq S_1$. Finally, our set is convex.

- Let $x_1, x_2 \in \{x \mid \exists y \in S_2, x + y \in S_1\}$. Then let y_1, y_2 be associated points to x_1, x_2 . For $x = \lambda x_1 + (1 - \lambda) x_2$ and $y = \lambda y_1 + (1 - \lambda) y_2$. Then, since S_2 is convex, $y \in S_2$. Moreover :

$$x + y = \lambda \underbrace{(x_1 + y_1)}_{\in S_1} + (1 - \lambda) \underbrace{(x_2 + y_2)}_{\in S_1} \\ \underbrace{\hspace{10em}}_{\in S_1 \text{ since } S_1 \text{ is convex}}$$

Then, our set is convex.

Convex Functions

- The hessian of f at (x, y) is the matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. On \mathbb{R}^2 this matrix is not positive nor negative and thus the function is neither convex, or concave. However, the upper level sets for α are convex if and only if $\alpha \geq 0$ (from our first example) and thus the function is neither quasi-concave nor quasi-convex.
- On \mathbb{R}_{+*}^2 , the hessian of the function is positive semidefinite and thus the function is convex.
- The hessian matrix of f is the matrix $\begin{pmatrix} 0 & -\frac{1}{x_2^2} \\ -\frac{1}{x_2^2} & \frac{2x_1}{x_2^3} \end{pmatrix}$. Since its determinant is < 0 the Hessian is not positive semi-definite and the function is not convex. However, its sublevel sets are defined by the equations $x_1 \leq \alpha x_2$ and are thus convex (since half-planes). Thus, f is quasi-convex.

- Let us define the Löwner order \preceq as the partial order defined by the convex cone of positive semi-definite matrices. We know that :

$$A \preceq B \Rightarrow B^{-1} \preceq A^{-1}, \forall A, B \in S_{++}^n$$

From this, we know that for all $t \leq 1$:

$$((1-t)X + tY)^{-1} \preceq (1-t)X^{-1} + tY^{-1}$$

By linearity of the trace, we can now see that $\boxed{X \mapsto \text{Tr}(X^{-1}) \text{ is convex}}.$

Fenchel Conjugate

- We have :

$$\boxed{f^*(y) = \sup_x (\langle xy \rangle - \|x\|) = \begin{cases} 0 & \text{if } \|y\|_* \leq 1 \\ \infty & \text{otherwise} \end{cases}}$$

To show this, let y such that $\|y\|_* = \sup_{\|x\| \leq 1} \langle xy \rangle \leq 1$. Then by Cauchy-Schwarz inequality, we know that $\langle xy \rangle \leq \|x\| \|y\|_* \leq \|x\|$ and the term in the supremum is always ≤ 0 . If $\|y\|_* > 1$ however, then there exists z such that $\|z\| \leq 1$ such that $\langle zy \rangle > 1$. Taking $x = tz$ in the supremum, we know that $f^*(y) \geq t(\langle zy \rangle - \|z\|)$ which goes to infinity with $t \rightarrow \infty$. Thus, the fenchel conjugate of a norm is the convex indicator of the unit ball of the dual norm.

- Let us denote the infimal convolution of g, h by $g \square h$. Then we will show that :

$$\boxed{(g \square h)^* = (g^* + h^*)}$$

Note that :

$$\begin{aligned} (g \square h)^*(\alpha) &= \sup_{x,y} \{ \langle x\alpha \rangle - g(y) - h(x-y) \} \\ &= \sup_{x_1, x_2} \{ \langle (x_1 + x_2)\alpha \rangle - g(x_1) - h(x_2) \} \\ &= \sup_{x_1, x_2} \{ \langle x_1\alpha \rangle - g(x_1) \} + \sup_{x_2} \{ \langle x_2\alpha \rangle - h(x_2) \} \\ &= g^* + h^* \end{aligned}$$

Given $g = \|\cdot\|_1$ and $h = \frac{1}{2\alpha} \|\cdot\|_2^2$. Let $x = u + v$. Then :

$$f(x) = \inf_v \{ g(x-v) + h(v) \}$$

Substituting the expressions, we get :

$$\begin{aligned} f(x) &= \inf_v \left\{ \|x-v\|_1 + \frac{1}{2\alpha} \|v\|_2^2 \right\} \\ &= \sum_{i=1}^n \inf_{v_i} \left\{ |x_i - v_i| + \frac{1}{2\alpha} v_i^2 \right\} \end{aligned}$$

We will now compute the infima independently. We have two cases :

1. $v_i \leq x_i$: Let $\varphi(v_i) = x_i - v_i + \frac{1}{2\alpha} v_i^2$. We want to find a minimum for φ , which is found at $v_i = \alpha$. This solution is valid if $\alpha \leq x_i$.
2. $v_i > x_i$: Let $\varphi(v_i) = \frac{1}{2\alpha} + v_i - x_i$. We want to find a minimum for φ , which is at $v_i = -\alpha$. This solution is valid if $\alpha > x_i$.

Then, we have three possible cases for the optimal v_i :

1. If $x_i \geq \alpha$, $v_i = \alpha$
2. If $x_i \leq -\alpha$, $v_i = -\alpha$
3. If $-\alpha < x_i < \alpha$, $v_i = x_i$.

Plugging this into f :

$$f(x) = \sum_{i=1}^n \left(\begin{cases} \frac{1}{2\alpha} x_i^2 & |x_i| \leq \alpha \\ |x_i - \alpha| - \frac{\alpha}{2} & |x_i| > \alpha \end{cases} \right)$$

Now, clearly, $f = g \square h$ is a convex function (as the infimum of two convex functions (if $\alpha > 0$)). Moreover, f is clearly lower semi-continuous from its expression. Now, we have $\boxed{f^{**} = f}$ from the Fenchel-Moreau theorem (proved below) :

Théorème 1.1 The biconjugate of f is the largest lower semi-continuous convex function below than f .

Démonstration. Let $x \in \mathbb{R}^n$. For all y , the directional derivative :

$$\partial_y f(x) = \lim_{dt \rightarrow 0} \frac{f(x + dt y) - f(x)}{dt}$$

is a sublinear as a function of y . From the Hahn-Banach theorem, there exists $\tilde{\partial} f \in (\mathbb{R}^n)^* = \mathbb{R}^n$ such that :

$$\langle \tilde{\partial} f, \cdot \rangle \leq \partial_y f(x) \leq f(x + y) - f(x), \forall y \in \mathbb{R}^n$$

Then :

$$f(x) + f^*(\tilde{\partial} f) = \langle \tilde{\partial} f, x \rangle$$

which completes our proof that $f^{**} = f$. ■

- We want to compute :

$$\ell^*(y) = \sup_{z \in \mathbb{R}} \{yz - \ell(z)\} \text{ where } \ell : z \mapsto \log(1 + e^z)$$

We differentiate $\varphi(z)$ the function in the supremum :

$$\varphi'(z) = y - \frac{1}{1 + e^z}$$

Then, we find $z = \log\left(\frac{1-y}{y}\right)$, which only makes sense for $y \in]0, 1[$. Finally, we compute :

$$\begin{aligned} \ell^*(y) &= y \log\left(\frac{1-y}{y}\right) + \log(y) \\ &= y \log(1-y) - y \log(y) + \log(y) \\ &= y \log(1-y) + (1-y) \log(y) \end{aligned}$$

In the end,

$$\boxed{y \log(1-y) + (1-y) \log(y)}$$

Duality

- The Lagrangian of the problem associated with $Ax = b$ is :

$$\mathcal{L}(x, \lambda) = \frac{1}{2} \|Ax - b\|_2^2 + \alpha \|x\|_1 - {}^t\lambda (Ax - b) = \frac{1}{2} \|Ax - b\|_2^2 + \alpha \|x\|_1 - {}^t\lambda Ax + {}^t\lambda b$$

Then, by separating terms involving x :

$$\mathcal{L}(x, \lambda) = \frac{1}{2} \|b\|_2^2 - \frac{1}{2} \|\lambda - b\|_2^2 + {}^t\lambda b + \alpha \|x\|_1 - ({}^t\lambda A) x$$

In solving $\max_{\lambda} \min_x \mathcal{L}(x, \lambda)$, the inner minimization on x only depends on $\|x\|_1$ and $({}^t\lambda A) x$. Minimizing it results in $\|{}^t\lambda A\|_{\infty} \leq \alpha$. Finally, we get that :

$$\boxed{\max_{\lambda \in \mathbb{R}^m} -\frac{1}{2} \|\lambda - b\|_2^2 + \frac{1}{2} \|b\|_2^2 - \inf_{\{\|\cdot\|_{\infty} \leq 1\}} \left(\frac{{}^t\lambda A}{\alpha} \right) \text{ is a dual problem for the LASSO}}$$

- We start with :

$$\min_{w, w_1, \dots, w_m \in \mathbb{R}^n} \left\{ \frac{1}{n} \sum_{i=1}^m h_i(w_i) + \frac{\lambda}{2} \|w\|_2^2 \mid w_i = w, \forall i \in \llbracket 1, m \rrbracket \right\}$$

Its Lagrangian is :

$$\mathcal{L}(w, w_1, \dots, w_m, v_1, \dots, v_m) = \frac{1}{n} \sum_{i=1}^m h_i(w_i) + \frac{\lambda}{2} \|w\|_2^2 + \sum_{i=1}^m {}^t v_i (w - w_i) = \frac{\lambda}{2} \|w\|_2^2 + \sum_{i=1}^m {}^t v_i w + \sum_{i=1}^m \frac{1}{n} h_i(w_i) - {}^t v_i w_i$$

For the minimization step, we minimize for each of the w_i and for w . Let $g_i(v_i)$ be the minimized value of $\frac{1}{n} h_i(w_i) - {}^t v_i w_i$. For w , we want to minimize $\frac{\lambda}{2} \|w\|_2^2 + \sum_{i=1}^m {}^t v_i w$, which is quadratic in w . Computing its minimum we find $\frac{-1}{2\lambda} \|\sum_{i=1}^m v_i\|_2^2$. Finally, the dual problem is :

$$\boxed{\max_{v_1, \dots, v_m} \sum_{i=1}^m g_i(v_i) - \frac{1}{2\lambda} \left\| \sum_{i=1}^m v_i \right\|_2^2}$$

For logistic regression, we have :

$$h_i : w \mapsto \log(1 + \exp(-y_i {}^t x_i w))$$

Then we have :

$$g_i(v_i) = \min_{w_i} \left\{ \frac{1}{n} \log(1 + \exp(-y_i {}^t x_i w_i)) - {}^t v_i w_i \right\}$$

Since h_i is differentiable, we can compute its gradient :

$$\nabla h_i(w_i) = \frac{1}{n} \cdot \frac{-y_i x_i}{1 + \exp(y_i {}^t x_i w_i)}$$

which gives us a minimality condition :

$$\frac{-y_i x_i}{1 + \exp(y_i {}^t x_i w_i)} = n v_i$$

which rearranges to :

$$w_i = \frac{-y_i \log\left(\frac{-y_i {}^t v_i x_i - n \|v_i\|_2^2}{n \|v_i\|_2^2}\right)}{\|x_i\|_2^2} x_i = \frac{-y_i {}^t v_i x_i}{\|x_i\|_2^2} x_i$$

Inputing this into $y_i^t x_i w_i$:

$$y_i^t x_i w_i = + \underbrace{y_i^2}_{=1} \frac{\overbrace{x_i^t x_i}^{=1}}{\|x_i\|_2^2} t = t$$

Thus :

$$g_i(w_i) = \frac{1}{n} \log \left(-\frac{y_i^t v_i x_i}{\|v_i\|_2^2} \right) - \frac{y_i^t v_i w_i}{\|x_i\|_2^2} \log \left(-1 - \frac{y_i^t v_i x_i}{\|v_i\|_2^2} \right)$$

Finally our dual problem is :

$$\max_{v_1, \dots, v_m} \sum_{i=1}^m \frac{1}{n} \log \left(-\frac{y_i^t v_i x_i}{\|v_i\|_2^2} \right) - \frac{y_i^t v_i w_i}{\|x_i\|_2^2} \log \left(-1 - \frac{y_i^t v_i x_i}{\|v_i\|_2^2} \right) - \frac{1}{2\lambda} \left\| \sum_{i=1}^m v_i \right\|_2^2$$

and I will not be trying to find a closer form manually.

- We consider the problem :

$$\min_{X \in \mathbb{S}^n} \{ \text{Tr}(A_0 X) \mid X \succeq 0, \text{Tr}(A_1 X) = b_1, \dots, \text{Tr}(A_m X) = b_m \}$$

Its Lagrangian is (for $\lambda_i > 0, S \succeq 0$) :

$$\mathcal{L}(X, \lambda_1, \dots, \lambda_m, S) = \text{Tr}(A_0 X) + \sum_{i=1}^m \lambda_i (\text{Tr}(A_i X) - b_i) + \text{Tr}(S X) = \text{Tr} \left(\left(A_0 + \sum_{i=1}^m \lambda_i A_i + S \right) X \right) - \sum_{i=1}^m \lambda_i b_i$$

Minimizing everything in X can be done since everything is differentiable in X :

$$\nabla \mathcal{L} = {}^t A_0 + \sum_{i=1}^m \lambda_i {}^t A_i + 2S X = A_0 + \sum_{i=1}^m \lambda_i A_i + S$$

which is 0 when :

$$S = - \left(A_0 + \sum_{i=1}^m \lambda_i A_i \right)$$

Finally we get the dual problem :

$$\max_{\lambda_1, \dots, \lambda_m, S} \left\{ - \sum_{i=1}^m \lambda_i b_i \mid S \succeq 0, S = -A_0 - \sum_{i=1}^m \lambda_i A_i \right\}$$

Strong duality holds if, and only if, the Karush-Kuhn-Tucker conditions are satisfied. Since $\text{Tr}(S_* X_*)$ is a rewriting of complementary slackness, clearly strong duality implies $\text{Tr}(S_* X_*) = 0$. If we have $\text{Tr}(S_* X_*) = 0$ for the optimal solutions, in particular, both the primal and dual problems are attained, the gradient computed earlier vanishes and thus the KKT conditions are satisfied. Finally :

$$\boxed{\text{Strong Duality holds if and only if } \text{Tr}(X_* S_*) = 0}$$

2 Problem

We consider the following problem :

$$\min_{w \in \mathbb{R}^n} \frac{1}{2} \|w\|_2^2 + \alpha \sum_{i=1}^m \max \{0, 1 - {}^t w x_i y_i\} \quad (1)$$

2.1 Dual problem.

Considering the constraints, we have :

$$s_i \geq 1 - y_i {}^t w x_i \wedge s_i \geq 0$$

and thus, minimizing in s , for w fixed gives :

$$s_i = \max \{0, 1 - y_i {}^t w x_i\}$$

which proves the equivalent reformulation :

$$\begin{aligned} \min_{w \in \mathbb{R}^n, s \in \mathbb{R}^m} \quad & \frac{1}{2} \|w\|_2^2 + \alpha \sum_{i=1}^m s_i \\ \text{s. t.} \quad & {}^t w x_i y_i \geq 1 - s_i \\ & s_i \geq 0 \end{aligned}$$

Computing the dual problem needs two constraints :

$$\lambda_i \geq 0 \text{ for } {}^t w X_i \geq 1 - s_i \text{ and } \mu_i \geq 0 \text{ for } s_i \geq 0$$

The Lagrangian is thus :

$$\mathcal{L}(w, s, \lambda, \mu) = \frac{1}{2} \|w\|_2^2 + \alpha \sum_{i=1}^m s_i - \sum_{i=1}^m \lambda_i ({}^t w X_i - 1 + s_i) - \sum_{i=1}^m \mu_i s_i$$

Then :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} &= w - \sum_{i=1}^m \lambda_i X_i = 0 \Rightarrow w = X \lambda \\ \frac{\partial \mathcal{L}}{\partial s_i} &= \alpha - \lambda_i - \mu_i = 0 \Rightarrow \lambda_i \leq \alpha \end{aligned}$$

Finally, substituting, the dual problem is :

$$\max_{0 \leq \lambda \leq \alpha} \mathcal{D}(\lambda) = -\frac{1}{2} {}^t \lambda^t X X \lambda + \sum_{i=1}^m \lambda_i$$

2.2 Algorithms

Everything in this section has been implemented using Python 3.12. Preamble :

```

import functools

from random import randint
import numpy as np
import numpy.linalg as npl

def projection(lower, upper, value):
    return min(upper, max(value, lower))

def d(lam, X):
    return -(1/2) * lam.T.dot(X.T).dot(X).dot(lam) + sum(lam)

def gradientD(lam, X):
    return np.array([1 for _ in lam]) - X.T.dot(X).dot(lam)

```

- Implementation for Gradient Ascent

```

def gradient_ascent_dual_svm(dataset, alpha, iterations):
    """
    :param alpha: parameter of the SVM problem
    :param dataset: List of couples x_{i}:np.array, y_{i}:+-1 representing our classes
    :param iterations: Number of iterations to compute
    :return:
    """

    proj = functools.partial(projection, lower=0.0, upper=alpha)
    X = np.zeros((n:=len(dataset[0][0]), m:=len(dataset)))
    for i in range(m):
        X[i] = dataset[i][0] * dataset[i][1]

    lam = 0
    h = max(npl.eigvalsh(X))
    for _ in range(iterations):
        olam = lam
        lam = map(proj, olam + h * gradientD(olam, X))
        while d(lam, X) > d(olam, X) + (1/(h * 2)) * (npl.norm(lam - olam) ** 2):
            h = h / 2
            lam = map(proj, olam + h * gradientD(olam, X))

    return X.dot(lam)

```

- Implementation for Randomized Coordinate Ascent

```

def randomized_coordinate_ascent(dataset, alpha, iterations):
    proj = functools.partial(projection, lower=0.0, upper=alpha)
    lam = np.zeros(m:= len(dataset))
    n = len(dataset[0][0])
    w = np.zeros(n)
    for _ in range(iterations):
        i = randint(0, m - 1)
        olam = lam[i]
        lam = lam.copy()
        lam[i] = proj(olam + (1 - (yi := dataset[i][1]) * (xi := dataset[i][0]).T.dot(w))/(npl.norm(xi)**2))
        w = w + yi * xi.dot(lam[i] - olam)
    return w

```