
COMP0083: Advanced Topics in Machine Learning 2024/2025
Introduction to Convex Optimization

Coursework

Due Date: **3 January 2025**
Instructor: Massimiliano Pontil
TA: Riccardo Grazzi

Instructions There are 3 sections. Two assess the knowledge of the theory and one requires implementing algorithms. Provide sufficient explanations for all solutions. For the coding part, you have to provide python code relying on the following libraries: `numpy`, `scipy`, `sklearn`, `matplotlib`.

Deadlines Submit your report by January 3 2025.

Policies **Late submissions:** past the online due date, late submissions will be penalized by 20% of the original total score per late day (e.g., 40% for 2 days). Excused extensions will be given only for significant issues and if requested well in advance the due date.

1 Questions with multiple answers [30%]

Question 1 [6%] Which one of these is a convex function? ($a, b \in \mathbb{R}$)

a) $f(x) = \max\{ax + b, x^4 - 5, e^{-x}\}$

b) $g(x) = \min\{ax + b, x^4 - 5, e^x\}$

c) $h(x) = x^4 - 5 - e^x$.

Question 2 [8%] Consider the function

$$f(x) = \begin{cases} -x & \text{if } x \in]-1, 0] \\ x^2 & \text{if } x \geq 0. \end{cases}$$

Indicate which one of the two graphs below represents the subdifferential of f .

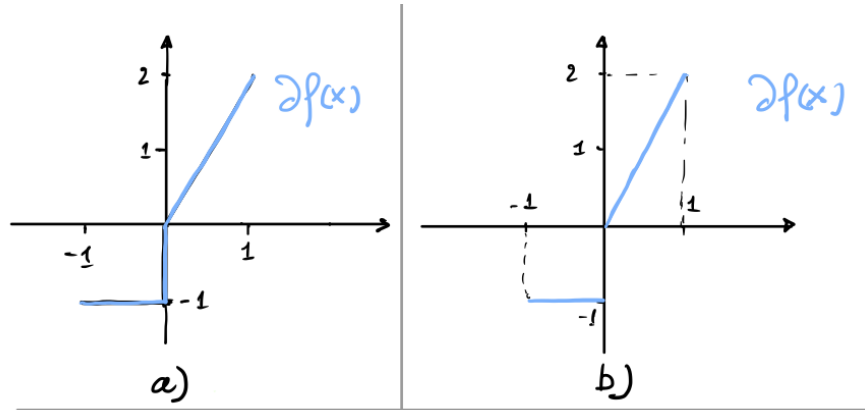


Figure 1: Question 2

Question 3 [8%] Let A is a square matrix, not necessarily symmetric and A^* be the transpose of A . Let $f(x) = \langle Ax, x \rangle + \langle x, b \rangle + c$. Which of the following is the gradient of f ?

- a) $A^*x + Ax + b$
- b) $2Ax + b$
- c) $A^* + b$

Question 4 [8%] Let $g \in \Gamma_0(\mathbb{R})$. The Fenchel conjugate of $f(x) = g(3x)$ is

- a) $f^*(u) = g^*(u/3)$
- b) $f^*(u) = g^*(3u)$
- c) $f^*(u) = g^*(u)$.

2 Theory on convex analysis and optimization [40%]

Problem 1. [9%] Compute (showing a complete derivation) the Fenchel conjugates of the following functions.

1. $f: \mathbb{R} \rightarrow]-\infty, +\infty]$, with $f(x) = \begin{cases} +\infty & \text{if } x \leq 0 \\ -\log x & \text{if } x > 0 \end{cases}$.
2. $f(x) = 2x^2$.
3. $\iota_{[-1,1]}$ (that is, the function which is zero on the interval $[-1,1]$, and $+\infty$ otherwise).

Problem 2. [9%] Let $f: X \rightarrow]-\infty, +\infty]$ be a proper convex function.

1. Prove by induction the Jensen's inequality, that is

$$\sum_{i=1}^n f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i),$$

for all $x_1, \dots, x_n \in X$ and for all $\lambda_1, \dots, \lambda_n \in \mathbb{R}_+$ with $\sum_{i=1}^n \lambda_i = 1$.

2. Using the characterization for differentiable functions prove that the function $-\log$ is convex
3. Applying Jensen's inequality to $-\log$, prove the following arithmetic-geometric inequality

$$\sqrt[n]{x_1 \cdots x_n} \leq \frac{1}{n} \sum_{i=1}^n x_i,$$

for all $x_1, \dots, x_n \in \mathbb{R}^+$.

Problem 3. [5%] Given a polytope $C = \text{co}(a_1, \dots, a_m)$ in X , where, recall, “co” denotes the convex hull. Prove that the maximum of a convex function f on C is attained at one of the vertices a_1, \dots, a_m .

Problem 4. [5%] Let $f: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, with $f(x, y) = \|2x - y\|_2^2$. Prove that f is (jointly) convex.

Problem 5. [12%] Consider the optimization problem.

$$\min_{x \in \mathcal{C}} \frac{1}{2} \|x\|^2, \quad \mathcal{C} = \{x : \|Ax - b\|_\infty \leq \varepsilon\}$$

where $\varepsilon > 0$, $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$. Solve the following points.

1. Compute the dual problem (using the Fenchel-Rockafellar duality theory). [\[Hint: put the problem in the form \$f\(x\) + g\(Ax\)\$ \]](#)
2. Does strong duality hold? Justify the answer. [\[Hint: use the qualification conditions\]](#)
3. Write the KKT conditions.
4. Derive a rate of convergence on the primal iterates from the application of FISTA on the dual problem. [\[Hint: recall that it is possible to bound the square of the distance to the primal solution by the dual objective values\]](#)

3 Solving the Lasso problem [30%]

Consider the following problem

$$\min_{x \in \mathbb{R}^d} \frac{1}{2n} \|Ax - y\|_2^2 + \lambda \|x\|_1, \quad (1)$$

where $A \in \mathbb{R}^{n \times d}$. Let us denote by $a^i \in \mathbb{R}^d$ and $a_j \in \mathbb{R}^n$ the i -th row and j -column of A respectively. Then the objective function can be written in the form

$$\frac{1}{2n} \sum_{i=1}^n (\langle a^i, x \rangle - y_i)^2 + \lambda \|x\|_1, \quad (2)$$

Then the *proximal stochastic gradient algorithm* is

$$x^{k+1} = \text{prox}_{\gamma_k \lambda \|\cdot\|_1} (x^k - \gamma_k (\langle a^{i_k}, x \rangle - y_{i_k}) a^{i_k}), \quad (\text{PSGA})$$

where $\gamma_k = n/(\|A\|_2^2 \sqrt{k+1})$ and $(i_k)_{k \in \mathbb{N}}$ is a sequence of i.i.d. random variables uniformly distributed on $\{1, \dots, n\}$.

Alternatively, problem (1) can be approached via a *randomized coordinate proximal gradient algorithm*. Indeed, recalling that $\frac{d}{dx} (1/2) \|Ax - y\|_2^2 = A^*(Ax - y) = (\langle a_j, Ax - y \rangle)_{1 \leq j \leq d}$ one can consider the following algorithm

$$x_j^{k+1} = \begin{cases} \text{prox}_{\gamma_j \lambda |\cdot|} (x_j^k - \frac{\gamma_j}{n} \langle a_j, Ax^k - y \rangle) & \text{if } j = j_k \\ x_j^k & \text{otherwise,} \end{cases} \quad (\text{RCPGA})$$

where $(j_k)_{k \in \mathbb{N}}$ is a sequence of i.i.d. random variables uniformly distributed on $\{1, \dots, d\}$ and $\gamma_j = n/\|a_j\|_2^2$.

Generate the data according to the following python code with $n = 1000$, $d = 500$, $s = 50$, $\sigma = 0.06$.

```
def generate_problem(n, d, s, std=0.06):
    # Generate xs
    # vectors with entries in [0.5, 1] and [-1, -0.5]
    # respectively
    assert s % 2 == 0, "s needs to be divisible by 2"
    xsp = 0.5 * (np.random.rand(s // 2) + 1)
    xsn = - 0.5 * (np.random.rand(s // 2) + 1)
    xsparse = np.hstack([xsp, xsn, np.zeros(d - s)])
    random.shuffle(xsparse)

    # Generate A
    A = np.random.randn(n, d)
```

```

# Generate eps
y = A @ xsparse + std * np.random.randn(n)

return xsparse, A, y

```

1. Implement algorithm (PSGA), recalling that $\text{prox}_{\gamma_k \lambda \|\cdot\|_1}$ acts component-wise as a soft-thresholding operator with threshold $\gamma_k \lambda$.
2. Implement algorithm (RCPGA).
3. Choose an appropriate regularization parameter λ and plot the objective function values vs the number of iterations for both the algorithms described above. For algorithm (PSGA) consider also the behavior of the objective values over the sequence of ergodic means, that is, $\bar{x}^k = (\sum_{i=0}^k \gamma_i)^{-1} \sum_{i=0}^k \gamma_i x_i$.