Tutorial 4: Regularization

Optimization for data science, M2 MIAGE ID/ID Apprentissage

January 21, 2026

Dauphine | PSL
UNIVERSITÉ PARIS

Exercise 1: Elastic net

We consider a dataset formed by $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} \in \mathbb{R}^n$. Given this dataset, we form a linear regression problem with so-called *elastic net* regularization :

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimize}} \frac{1}{2n}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}\|^2 + \lambda_1\|\boldsymbol{w}\|_1, \tag{1}$$

where $\lambda_2 \geq 0$ and $\lambda_1 \geq 0$.

a) What is the role of a regularization term in general?

b) What is the purpose of the regularization term when $\lambda_1 = 0$ and $\lambda_2 > 0$?

c) What is the purpose of the regularization when $\lambda_2 = 0$ and $\lambda_1 > 0$?

d) Recall that the gradient of the function $\phi : \boldsymbol{w} \mapsto \frac{1}{2n}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$ is given by

$$\nabla\phi(\boldsymbol{w}) = \frac{1}{n}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}).$$

Using this formula, write down the iteration of proximal gradient for problem (1).

e) When $\lambda_2 = 0$ and $\lambda_1 > 0$, which algorithm is proximal gradient equivalent to?

f) When $\lambda_1 > 0$ and $\lambda_2 > 0$, there does not exist an explicit formula for the proximal gradient iterates, and the proximal subproblem has to be solved approximately at every iteration. Propose an algorithm among those seen in class that could be employed to compute such an approximate solution, and justify your choice of that particular method.

Exercise 2: Reversed Huber loss

In this exercise, we consider the reverse philosophy of the Huber loss, that is, we propose to use a loss function that looks like the absolute value on $[-1, 1]$ and like a quadratic everywhere else. The *reversed Huber loss* is thus defined as:

$$\begin{array}{rccl} r: & \mathbb{R} & \to & \mathbb{R} \\ & t & \mapsto & r(t) := \begin{cases} |t| & \text{if } |t| < 1 \\ \frac{t^2+1}{2} & \text{otherwise.} \end{cases} \end{array} \tag{2}$$

This function is convex but nonsmooth, since it is not differentiable at 0.

As in Exercise 1, we consider linear models $\boldsymbol{x} \mapsto \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ and a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$.

a) We first consider the convex, nonsmooth problem:

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} r(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i). \tag{3}$$

i) What mathematical tool can we use to design algorithms applicable to problem (3)?

ii) Using this tool, how can the solutions of (3) be characterized?

b) We now study the family of problems:

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} f(\boldsymbol{w}) + \lambda\sum_{i=1}^{d} r([\boldsymbol{w}]_i), \tag{4}$$

where $f(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{w})$ with every f_i $\mathcal{C}^1$ and depending on the data point $(\boldsymbol{x}_i, y_i)$, and $\lambda > 0$.

i) How is this type of problem called? What is the purpose of the second term?

ii) Write the generic proximal gradient iteration for this problem.

iii) When is this algorithm worthy of consideration in practice?

Solutions

Solutions for Exercise 1

a) A regularization term enforces a desired structure on the optimization variables.

b) When $\lambda_1 = 0$ and $\lambda_2 > 0$, the regularization term is an ℓ_2 regularization term, that aims at reducing the variance of the solution with respect to the data.

c) When $\lambda_2 = 0$ and $\lambda_1 > 0$, the regularization term is an ℓ_1 regularization term, that aims at promoting sparse solutions.

d) The kth iteration of proximal gradient applied to problem (1) is

$$\boldsymbol{w}_{k+1} \in \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \phi(\boldsymbol{w}_k) + \frac{1}{n}(\boldsymbol{X}\boldsymbol{w}_k - \boldsymbol{y})^{\mathrm{T}}\boldsymbol{X}(\boldsymbol{w} - \boldsymbol{w}_k) + \frac{1}{2\alpha_k}\|\boldsymbol{w} - \boldsymbol{w}_k\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}\|^2 + \lambda_1\|\boldsymbol{w}\|_1 \right\}$$

where $\alpha_k > 0$.

e) A subgradient algorithm can be used to solve the subproblem, since it is defined even in the presence of a nonsmooth function such as the ℓ_1 norm. The iteration cost would involve computing a subgradient. Alternatively, one may want to apply proximal gradient to this subproblem while treating the ℓ_1 norm as a regularization term. This would correspond to the ISTA method (with a quadratic cost function), and would still be tractable given that the iterations of ISTA are explicitly defined.

Solutions for Exercise 2

a)

i) Since $\boldsymbol{w} \mapsto \frac{1}{n}\sum_{i=1}^n r(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)$ is convex, it is possible to define the subdifferential of v at any point: the elements of the subdifferential, called the subgradients, can be used in lieu of the gradient to construct optimization methods for solving problem (3).

ii) Let $\phi_r : \boldsymbol{w} \mapsto \frac{1}{n}\sum_{i=1}^n r(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)$. A point $\bar{\boldsymbol{w}} \in \mathbb{R}^d$ is a global minimum of ϕ_r if and only if

$$\mathbf{0} \in \partial\phi_r(\bar{\boldsymbol{w}}),$$

where $\partial\phi_r(\cdot)$ denotes the subdifferential of ϕ_r.

b)

i) Problem (4) is a regularized optimization problem. The goal of the second term, that does not depend on data, is to enforce desired properties for the solution.

ii) At a point $\boldsymbol{w}_k$, the generic proximal gradient iteration (with a generic stepsize α_k) for this problem is:

$$\boldsymbol{w}_{k+1} \in \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ f(\boldsymbol{w}_k) + \nabla f(\boldsymbol{w}_k)^{\mathrm{T}}(\boldsymbol{w} - \boldsymbol{w}_k) + \frac{1}{2\alpha_k}\|\boldsymbol{w} - \boldsymbol{w}_k\|_2^2 + \lambda\sum_{i=1}^d r([\boldsymbol{w}]_i) \right\}.$$

iii) The proximal gradient algorithm is only interesting when the cost of solving the subproblem is cheaper than that of solving the original problem.