2.2 Binary Classification Problem

Given an input-output space $\mathcal{A} \times \{-1, +1\}$ we want to find a *prediction function* that given an input $a \in \mathcal{A}$ returns an accurate prediction of the true output b associated with a.
We proceed as follows. First of all we define a parametrized family of models, that is, a family of probability distributions $\mathbb{P}_x$ such that $\mathbb{P}_x(b|a)$ denotes the probability that the correct output associated with a is b according to the model defined by the parameter $x \in \mathbb{R}^n$. Among the family, we choose the model $\mathbb{P}_{\bar{x}}$ that better represents the relationship between inputs and outputs, in the sense that we will soon specify. Once we have fixed the model, given an input $a \in \mathcal{A}$ we take as a prediction of its output the value $b \in \{-1, +1\}$ that maximizes $\mathbb{P}_{\bar{x}}(b|a)$.

Specifically, let us consider the case $\mathcal{A} = \mathbb{R}^n$ and let us assume that a set of i.i.d. samples $\{(a_j, b_j)\}_{j=1}^N$ (*training set*) is given. Moreover, we assume that the models of the family are all of the same form, namely

$$\mathbb{P}_x(b|a) := \frac{1}{1 + e^{-ba^t x}} \qquad \text{for every} \quad x \in \mathbb{R}^n.$$

We take the parameter $\bar{x}$ that maximizes the likelihood over the training set. That is the point $\bar{x} \in \mathbb{R}^n$ that realizes

$$\max_{\mathbb{R}^n} \mathbb{P}_x(b_1, \ldots, b_n | a_1, \ldots a_n) = \max_{\mathbb{R}^n} \prod_{j=1}^{N} \mathbb{P}_x(b_j | a_j).$$

By the definition of $\mathbb{P}_x$ and taking the logarithm, the above problem is equivalent to find $\bar{x}$ that realizes

$$\min_{\mathbb{R}^n} f_{\mathcal{N}}(x), \qquad f_{\mathcal{N}}(x) = \frac{1}{N} \sum_{j=1}^{N} \log\left(1 + \exp(-b_j a_j^t x)\right) \tag{2.10}$$

which, denoting with $f_1, \ldots, f_N$ the terms of the sum, is a problem of the form (2.1).

Given a model $\mathbb{P}_x$ and an input a we define the predicted output according to the model $\mathbb{P}_x$ as $\hat{b} = 1$ if

$$\frac{1}{1+e^{-a^t x}} \geq 0.5$$

and $\hat{b} = -1$ otherwise.

Remark 2.3. In practical applications a regularization parameter is usually added to the objective function in (2.10). The motivation is the need to avoid overfitting, that is, to prevent the model to approximate the training set too well and thus not be able to generalize when applied to inputs that are not in the initial sample. Specifically we consider a l_2 norm regularization, getting

$$\min_{\mathbb{R}^n} f_{\mathcal{N}}(x), \qquad f_{\mathcal{N}}(x) = \frac{1}{N}\sum_{j=1}^{N} \log\left(1+\exp(-b_j a_j^t x)\right) + \lambda\|x\|^2. \tag{2.11}$$

Once the approximation $\bar{x}$ of the minimizer of (2.11) is computed, the associated model $\mathbb{P}_{\bar{x}}$ goes through a validation process. That is, we need to ensure that $\mathbb{P}_{\bar{x}}$ is a good representation not only of the samples in the training set that we used to define the objective function, but also of the whole input-output space. In order to do so, we assume that another set $\{(\tilde{a}_j, \tilde{b}_j)\}_{j=1}^{N_V}$ (called *validation set*) of known samples is given, then we can proceed in two ways. A first option is to compute the testing error corresponding to $\bar{x}$. That is, we consider the analogous of problem (2.10) for the validation set

$$\min_{\mathbb{R}^n} \tilde{f}(x), \qquad \tilde{f}(x) = \frac{1}{N_V}\sum_{j=1}^{N_V} \log\left(1+\exp(-\tilde{b}_j \tilde{a}_j^t x)\right), \tag{2.12}$$

we compute the optimal gap in $\bar{x}$, given by $e_V := \tilde{f}(\bar{x}) - \tilde{f}^*$ and the model is accepted if e_V is maller than a fixed ϵ. The second option is to count the correct predictions over the samples in the validation set. For each input $\tilde{a}_j$ in the validation set we compute the predicted output $\hat{b}_j$ according to the model and we check if it correspond to the correct output $\tilde{b}_j$ associated to $\tilde{a}_j$. The model is then accepted if the percentage of correctly predicted outputs is higher than a fixed threshold.

If the model is rejected, the optimization process is repeated, choosing a different training set or a different value of the regularization parameter λ.

3.3 Implementation

In this section we will study the implementation of the methods that we introduced in section 3.1 when applied to binary classification problems that we introduced in section 2.2

$$\min_{\mathbb{R}^n} f_{\mathcal{N}}(x) = \frac{1}{N}\sum_{j=1}^{N} f_j(x), \qquad f_j(x) = \log\left(1 + \exp(-b_j a_j^t x)\right) + \lambda\|x\|^2. \tag{3.25}$$

Computing the gradient of f_j for a generic index j, we get

$$\nabla f_j(x) = -\frac{\exp(-b_j a_j^t x)}{1 + \exp(-b_j a_j^t x)} b_j a_j + 2\lambda x, \tag{3.26}$$

thus the evaluation of the gradient does not require any additional scalar products to be computed, with respect to the evaluation of the function. Relying on this remark, at the beginning of every iteration the values $\exp(-b_j a_j^t x_k)$ are computed and then exploited during the execution to evaluate both the sampled function $f_{\mathcal{N}_k}(x_k)$ and the sampled gradient $\nabla f_{\mathcal{N}_k}(x_k)$.
We will now present the algorithms corresponding to the different phases of the method, referring to Algorithm 3.1.

Input

We distinguish among three different kinds of input values.

- Problem parameters: N full sample size, $\{a_j, b_j\}_{j=1}^N$ data-set, λ regularization parameter.
- Method parameters:
  - $x_0 \in \mathbb{R}^n$ starting point,
  - N_0 initial sample size,
  - τ growing factor of the sizes sequence $\{N_k\}$,
  - $(i_M,\ i_S,\ i_L,\ i_Y)$ vector of parameters corresponding to the chosen method. Specifically we assume $i_M = 1$ for the Spectral Gradient and $i_M = 2$ for the Spectral Conjugate Gradient, $i_S = 1$ if the subsamples sequence is nested and $i_S = 0$ otherwise, i_L equals to 1 or 2 if the Li-Fukushima condition or the Grippo condition