Homework Assignment N°2

BML36
Thibault Douzon
Rajavarman Mathivanan

September 12th, 2018

Contents

1 Exercise 2: Logistic classification & discrimination

From now we will use $\sigma(x) = \frac{1}{1+e^{-x}}$

1.1 Part a

- ☐ Initialize w_0 ?
  1. Some fixed w_0 like $\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$
  2. Some computation around the dataset like the mean: $w_0 = \frac{1}{N}\sum_{i=1}^{N} x_i$, concatenated with a constant.
  3. Some random vector
- ☐ How to learn: for batch learning use this equation at each step

$$w_{n+1} = w_n - \eta \nabla E(w_n) = w_n - \eta \sum_{n=1}^{N} (y(n) - t_n)\, x_n$$

- ☐ How to stop the iterative process ?
  1. Stop when the norm of the difference vector is low: $\Delta_n = \frac{\|w_{n+1} - w_n\|}{\|w_n\|} < \epsilon$
     This is a commonly used criterion that stops the process when the steps we take are getting small compared to our current result.
  2. Stop after fixed number of iteration
     This ensures we won't enter in a infinite non-convergent process.
  3. Stop when a threshold error is reached: $E(w_n) < \epsilon$
     This is actually a bad idea because most of the time we can't be certain it is possible to reach such threshold on the error. It would result in an infinite process.

Our algorithm goes as follows:

1. Chose ϵ, N and η respectivelly for precision, maximum number of iterations and speed convergency.
2. Set current error Δ to $+\infty$ and n to 0
3. Chose the initial discriminant: $w_{current}$. WE NEED TO CHOSE THE METHOD !
4. While $\Delta > \epsilon \wedge n < N$ do
   (a) Compute and store next discriminant w_{next}:

$$w_{next} = w_{current} - \eta \sum_{n=1}^{N} \left(\sigma({w_{current}}^\top x_n) - t_n\right) x_n$$

(b) Compute and store the new error Δ:

$$\Delta = \frac{\|w_{next} - w_{current}\|}{\|w_{current}\|}$$

(c) Prepare for next iteration: store w_{next} in place of $w_{current}$ and increment n

5. If $\Delta > \epsilon$, it means we have not converged enough towards the limit. We should consider increasing N OR using another algorithm for convergence (eg. Newton-Raphson)

6. Result is stored in $w_{current}$, number of steps in n.

1.2 Part b

First important thing to notice is that the point $x = \begin{bmatrix} -1 & 1 \end{bmatrix}$ is missclassified. We define $\bar{x} = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}$ thus it means that when we compute $w^\top \bar{x}$ the sign of the result is incorrect.
In our case, $w^\top \bar{x} = 1$, thus the real class of x is 0.
The formula to update the weights is the following:

$$w_{new} = w - \eta\sigma(w^\top \bar{x})\bar{x}$$

We get the following result:

$$w_{new} \approx \begin{bmatrix} 0.5614 & 2.4386 & 1.5614 \end{bmatrix}$$

2 Exercise 3: Error function & Gradient descent

2.1 Part a

If $y_n = \sigma(w^\top x_n)$ then:

$$E(w) = \sum_{n=1}^{N} (t_n - y_n)^4$$

$$\nabla_w E(w) = -4 \sum_{n=1}^{N} (t_n - y_n)^3 (y_n(1 - y_n)) x_n$$

2.2 Part b

$$E(w) = \sum_{n=1}^{N} |t_n - y_n|$$

Computing the gradient of an absolute value implies to derivate an absolute value function which is not C^1. We won't be able to assign a value to the gradient if the value inside the absolute function is 0.

$$\nabla_w E(w) = \sum_{n=1}^{N} \begin{cases} -y_n(1 - y_n)x_n & \text{if } t_n - y_n > 0 \\ y_n(1 - y_n)x_n & \text{if } t_n - y_n < 0 \end{cases}$$