Homework 1
Mouselinos Spyridon
February 2020

Exersize 1

In logistic regression in the form of two-class case we assume that the log-ratio of the posteriors is a linear function of x.

The assumed model is thus:

$$\ln \frac{P(w_1|x)}{P(w_0|x)} = \theta^T x$$

We also know that because we have two classes $w_1, w_0 \rightarrow P(w_1|x) + P(w_0|x) = 1$

so if $\sigma(t) = \frac{1}{1+e^{-t}}$, then:

$$P(w_1|x) = \sigma(\theta^T x)$$

$$P(w_0|x) = 1 - \sigma(\theta^T x)$$

Exersize 1 (cont)

Now, given a set of Y training examples where $Y=(y_n, x_n)$ with $y_n \in \{0,1\}$ and $x_n \in \mathbb{R}^l$. If we assume statistical independence and try to solve the problem by ML we have:

$$\text{Likelihood}(\theta) = \text{jpdf}(P(y_n|x_n;\theta)) \Rightarrow$$

$$\Rightarrow \text{Log Likelihood}(\theta) = \ln(\text{jpdf}(P(y_n|x_n;\theta))$$

$$\Rightarrow L(\theta) = \ln \prod_{n=1}^{N} P(y_n|x_n;\theta) = [illegible]$$

$$= \ln \prod_{n=1}^{N} P(1|x_n;\theta)^{y_n} P(0|x_n;\theta)^{1-y_n}$$

Just like a Bernoulli Trial

$$\Rightarrow \ln \prod_{n=1}^{N} \sigma(\theta^T x_n)^{y_n} (1-\sigma(\theta^T x_n))^{1-y_n}$$

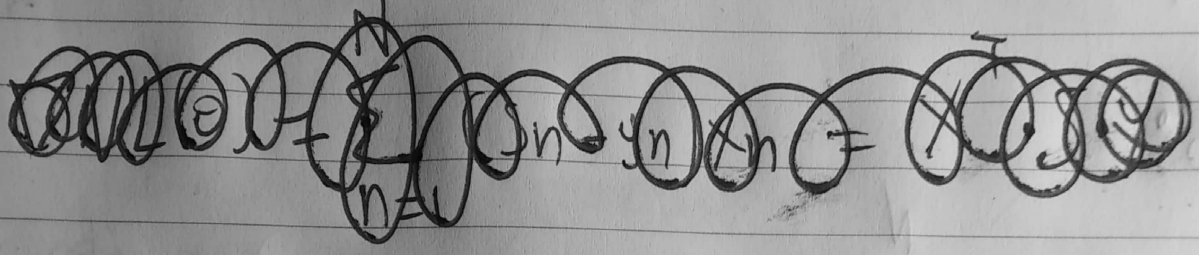
$$= \ln \prod_{n=1}^{N} s_n^{y_n} (1-s_n)^{1-y_n} \quad \text{where } s_n = \sigma(\theta^T x_n)$$

Exersize 1 (cont)

Now the negative log likelihood is

$$NL(\theta) = -\sum_{n=1}^{N} y_n \ln s_n + (1-y_n)\ln(1-s_n)$$

The gradient wrt θ is:



$$\nabla_\theta NL(\theta) = -\sum_{n=1}^{N} \frac{\partial}{\partial\theta}(y_n \ln s_n) + \frac{\partial}{\partial\theta}\left[(1-y_n)\ln(1-s_n)\right]$$

$$\nabla_\theta NL(\theta) = -\sum_{n=1}^{N} y_n \frac{\partial}{\partial\theta}\ln s_n + (1-y_n)\frac{\partial}{\partial\theta}\ln(1-s_n)$$

But:

$$\frac{\partial}{\partial\theta}\ln s_n = \frac{\partial}{\partial\theta}(s_n)\,\frac{1}{s_n} = \frac{1}{\sigma(\theta^T x)}\,\frac{\partial}{\partial\theta}\sigma(\theta^T x)$$

$$= \frac{1}{\sigma(\theta^T x)} \cdot \left(\sigma(\theta^T x)(1-\sigma(\theta^T x))x = (1-\sigma(\theta^T x))x\right.$$

In the same fashion:

$$\frac{\partial}{\partial\theta}\ln(1-s_n) = \frac{1}{1-s_n}\frac{d}{d\theta}(1-s_n)$$

$$= \frac{1}{1-\sigma(\theta^T x)}\left(-\sigma(\theta^T x)(1-\sigma(\theta^T x))x\right) = \left(-\sigma(\theta^T x)x\right)$$

So:

$$\nabla_\theta NL(\theta) = -\sum_{n=1}^{N}(1-\sigma(\theta^T x))y_n x_n + (1-y_n)\left(-\sigma(\theta^T x)x\right)$$

$$\nabla_\theta NL(\theta) = -\sum_{n=1}^{N} y_n x_n - y_n\sigma(\theta^T x)x_n + \left(-\sigma(\theta^T x)x\right.$$
$$\left. + \sigma(\theta^T x)x_n y_n\right)$$

$$\nabla_\theta NL(\theta) = -\sum_{n=1}^{N}(y_n - \sigma(\theta^T x))x_n = \sum_{n=1}^{N}\left(\sigma(\theta^T x) - y_n\right)$$

So finally:

$\nabla_\theta NL(\theta) = X^T(S-y)$, where

$X^T = [x_1, x_2, x_N]$, $S^T = [s_1, s_2, \ldots s_N]$

and $y^T = [y_1, y_2, \ldots y_N]$

A gradient update is given by:

$$\theta^{(\text{time step } i)} = \theta^{(\text{time step } i-1)} - \left(\mu_i \nabla NL(\theta)\right|_{\theta=\theta^{(i-1)}}$$

Meaning that:

$$\theta^t = \theta^{t-1} - \mu_i X^T(s^{t-1} - y)$$

μ_i is the learning rate parameter and t the time step of each update.

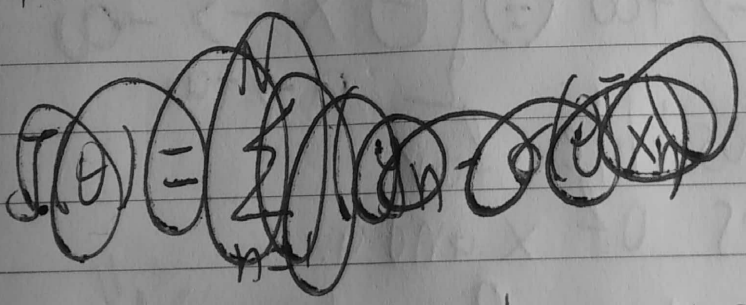
Exersize 2

a) Is in the HW8.ipynb.

b) As shown in Ex1. the gradient descent steep is:

$$\theta_j = \theta_{j-1} - \mu \nabla_\theta J(\theta)$$

where $J(\theta)$ is simply the cost function.



However here under MSE:

$$J(\theta) = \sum_{n=1}^{N} (y_n - f(\theta^T x))^2$$

So:

$$\nabla_\theta J(\theta) = -2 \sum_{n=1}^{N} (y_n - f(\theta^T x)) \cdot (f(\theta^T x_n)(1 - f(\theta^T x_n)) \cdot x_n \cdot a$$

Exersize 2 (cont)

So:

$$\theta_j = \theta_{j-1} + 2\mu \sum_{n=1}^{N} (y_n - f(\theta^T x)) \left(f(\theta^T x)(1 - f(\theta^T x)) \cdot a \right)$$

(c) In order for the model to respond with a clear 0 we should have $f(\theta^T x) = 0$

thus $\frac{1}{1 + e^{-\theta^T x} \cdot a} = 0$ that can happen

when $a \cdot e^{-\theta^T x} \to +\infty \Leftrightarrow \{ \theta^T x \to -\infty$ $a \to +\infty$

that is impossible for finite values of x and a.

the same applies for the "1" respon

Exersize 2 (cont)

(d) Having in mind the sigmoid function we can say that for a given set of (x, a) the model will respond with a value $f(\theta^T x)$. That value is the probability of the input to belong to class 0 or 1 on the threshold of 0.5, if $f(x^T\theta) > 0,5$ our model says class 1. if $f(x^T\theta) < 0,5$ class 0 respectivelly.

Exersize 2 (cont)

(e) A way of forcing our values close to 0 or 1 would be to use a large value of a as we showed on question (c).

Exersize 3

In this problem we are somewhat lucky, since the order in our dataset enables us to perform a "sort of" binary search in O(log(n)) in order to find the threshold.

We keep 3 pointers:

low ← The most negative point
mid ← The middle
high ← The most positive point

Iteratively we check if the mid and the high point have different classes (-1, +1). If they don't ~~the high point~~ the low is now the mid point. We keep until the low is at -1 the high at +1 and have convergence