Machine Learning
Exercise Sheet 2 Solution

Alper Hamza Ari 5771973 | Chenqi Hao 5781375 | Danil Skokov 5779466 | Said Orfan Haidari 5781295

November 1, 2023

1 Linear Regression

Please see `question1.ipynb` program for the solution to the given problem.

2 Logistic Regression

2.1 Deriving update rule for gradient descent

Given the objective loss function (negative loglikelihood or binary cross entropy loss):

$$J = -\sum_{i=1}^{N} y_n \log(p_i) + (1 - y_i) log(1 - p_i) \tag{1}$$

And predicted value of ith observation:

$$p_i = h_w(x_i) = P(y = 1|X = x_i; w) = \sigma(x_i) \tag{2}$$

We want to derive update rule of gradient descent optimization for logistic regression, which aims to find the optimal weights to minimize objective loss.
Given update rule and its parameters:

$$\begin{gathered} w^{(t+1)} = w^{(t)} - \eta \frac{\partial J(D, w)}{\partial w} \\ \text{where } t \in \mathbb{N}^+ \\ \text{and given dataset } D = \{(x_i, y_i)\}_{i=1}^{N} \end{gathered} \tag{3}$$

For simplicity, i will remove i from all equations and i will perform all the operations in matrix form.
Re-expressing what we had:

$$\begin{gathered} J = -ylog(p) - (1 - y)log(1 - p) \\ p = \sigma(z) = \frac{1}{1 + e^{-z}} \\ z = w^{\intercal} x + b \end{gathered} \tag{4}$$

By chain rule:

$$\begin{aligned} \frac{\partial J}{\partial w} &= \frac{\partial J}{\partial p} \cdot \frac{\partial p}{\partial w} \\ &= \frac{\partial J}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial w} \end{aligned} \tag{5}$$

And by calculating each partial derivative:

$$
\begin{aligned}
\frac{\partial J}{\partial p} &= \frac{-y}{p} + \frac{1-y}{1-p} \\
\frac{\partial p}{\partial z} &= \frac{e^{-z}}{(1+e^{-1})^2} \\
\frac{\partial z}{\partial w} &= x
\end{aligned} \tag{6}
$$

By putting pieces together:

$$
\frac{\partial J}{\partial w} = \left(\frac{-y}{p} + \frac{1-y}{1-p}\right)\left(\frac{-e^{-z}}{(1+e^{-z})^2}\right) \cdot (x) \tag{7}
$$

And recall that:

$$
\begin{aligned}
p &= \frac{1}{1+e^{-z}} \\
e^{-z} &= \frac{1-p}{p}
\end{aligned} \tag{8}
$$

Then our final partial derivative can be expressed again as follows:

$$
\begin{aligned}
\frac{\partial J}{\partial w} &= \left(\frac{-y}{p} + \frac{1-y}{1-p}\right) \cdot \left(\frac{(1-p)p^2}{p}\right) \cdot (x) \\
&= (p-y)x
\end{aligned} \tag{9}
$$

Which can also be expressed in different ways as follows:

$$
\begin{aligned}
\frac{\partial J}{\partial w} &= \sum_{i}^{N} -(y_i - h_w(x_i)) \cdot x_i \\
&= -X^\intercal (y-p)
\end{aligned} \tag{10}
$$

2.2 One step of gradient descent algorithm

We will append bias term to the end of weight array and define a new parameter array:

$$
w = \begin{bmatrix} w_1 & w_2 & b \end{bmatrix} \tag{11}
$$

which was initially given as:

$$
w = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{12}
$$

Then we will find predicted values for each observation x_n:

$$
\begin{gathered}
\text{Predicted value: } p = \sigma(w^\intercal x) \\
p_1(\text{or } \hat{y}_1) = \sigma((w^\intercal x_1)) = \sigma(0) = 0.5 \\
p_2(\text{or } \hat{y}_2) = \sigma((w^\intercal x_2)) = \sigma(0) = 0.5 \\
p_3(\text{or } \hat{y}_3) = \sigma((w^\intercal x_3)) = \sigma(0) = 0.5 \\
p_4(\text{or } \hat{y}_4) = \sigma((w^\intercal x_4)) = \sigma(0) = 0.5 \\
\text{which gives us the } \hat{y} \text{ array} \\
\hat{y} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}
\end{gathered} \tag{13}
$$

Then we will find gradient vector:

$$\frac{\partial J}{\partial w} = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \\ \frac{\partial J}{\partial b} \end{bmatrix} = \begin{bmatrix} (\hat{y} - y)x_1 \\ (\hat{y} - y)x_2 \\ 0 \end{bmatrix}$$

$$\begin{gathered}
\text{where } x_n \text{ represents array of observations for } nth \text{ feature} \\
x_1 = \begin{bmatrix} 2 \\ 3 \\ -4 \\ -2 \end{bmatrix} \\
x_2 = \begin{bmatrix} 4 \\ 3 \\ -2 \\ -6 \end{bmatrix} \\
\text{y represents array of results} \\
\text{and } \hat{y} \text{ represents array of predicted values} \\
y = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \\
\hat{y} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \\
\text{Then calculating gradient vector} \\
\frac{\partial J}{\partial w_1} = \begin{bmatrix} -0.5 & -0.5 & 0.5 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ -4 \\ -2 \end{bmatrix} = -5.5 \\
\frac{\partial J}{\partial w_2} = \begin{bmatrix} -0.5 & -0.5 & 0.5 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 3 \\ -2 \\ -6 \end{bmatrix} = -7.5 \\
\frac{\partial J}{\partial w} = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \\ \frac{\partial J}{\partial b} \end{bmatrix} = \begin{bmatrix} -5.5 \\ -7.5 \\ 0 \end{bmatrix}
\end{gathered} \tag{14}$$

Also notice that $\frac{\partial J}{\partial b} = 0$ since we neglect bias in the formula $z = w^\intercal x + b$ which leaves us with $z = w^\intercal x$. Please check equation 4 above.
Calculating new weights after 1 step:

$$w^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - (0.5) \begin{bmatrix} -5.5 \\ -7.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.75 \\ 3.75 \\ 0 \end{bmatrix} \tag{15}$$

Predicting $P(y = 1 | X = [-1, 1])$:

$$\begin{aligned}
\sigma(w^\intercal x + b) &= \sigma(w_1 x_1 + w_2 x_2 + b) \\
&= \sigma(-2.75 + 3.75 + 0) \\
&= \sigma(1) \\
&= \frac{1}{1 + e^{-1}} \\
&= 0.731
\end{aligned} \tag{16}$$