MLE 2: Exercise 4

May 18, 2025

I will reference all values of x_i with capital X for the ease of notation (usually for a sum).

1 Task 1

The Bernoulli distribution describes the distribution of binary variables, such as the outcomes of tossing a coin once. The positive outcome is usually denoted as "1", and the negative as "0", with probabilities $p(x)$ and $1 - p(x)$ respectively. The general formula is $p^x(1-p)^{1-x}$. We are looking to ensure that $\forall x_i, y_i \in D = \{(x_i, y_i)\}_{i=1}^N,\ p(x_i)^{y_i}(1-p(x_i))^{1-y_i}$.

The likelihood function for the Bernoulli distribution will look like this:

$$Ł(p(X)|X) = \prod_{i=1}^{N} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} = \prod_{i=1}^{N} p(x_i)^{y_i} \cdot \prod_{i=1}^{N}(1-p(x_i))^{1-y_i} = \tag{1}$$

$$= p(X)^{\sum_{i=1}^N y_i} \cdot (1-p(X))^{\sum_{i=1}^N (1-y_i)} = p(X)^{\sum_{i=1}^N y_i} \cdot (1-p(X))^{N-\sum_{i=1}^N y_i} = \tag{2}$$

$$= p(X)^Y \cdot (1-p(X))^{N-Y} \tag{3}$$

Log-likelihood, would, therefore, be the log of that expression:

$$l(p(X)|X) = log[Ł(p(X)|X)] = log[p(X)^Y] + log[(1-p(X))^{N-Y}] = \tag{4}$$

$$= Y \cdot log(p(X)) + (N-Y) \cdot log(1-p(X)) \tag{5}$$

We are looking for the parameters with which the loss function is minimal (i.e., $\hat{y}_i - y_i = 0$). This should be the case when the slope of the function approximates zero, so we need to derive the log-likelihood.

Derivation goes like this:

$$\frac{\partial\ l(p(X)|X)}{\partial\ p(X)} = \frac{\partial\ (Y \cdot log(p(X)) + (N-Y) \cdot log(1-p(X)))}{\partial\ p(X)} = \tag{6}$$

$$= Y \cdot \frac{1}{p(X)} - (N-Y) \cdot \frac{1}{1-p(X)} = 0 \tag{7}$$

Let's solve it:

$$\frac{Y}{p(X)} = \frac{N-Y}{1-p(X)} \tag{8}$$

$$\frac{1-p(X)}{p(X)} = \frac{N-Y}{Y} \tag{9}$$

$$\frac{1}{p(X)} - 1 = \frac{N}{Y} - 1 \tag{10}$$

$$p(X) = \frac{Y}{N} \tag{11}$$

We see that the probability of the desired answer under the derivative $= 0$ is the mean of the golden outputs. Given that $y_i \in \{0; 1\}, \forall y_i \in Y$, $\overline{Y} = 0,5$.

2 Task 2

2.1 Part (a)

Negative log-likelihood is a version of the mean squared error. It prevails over the standard formula (MSE) because it allows for simplification of the calculation. It is used as a loss function to guide the learning process.

Foremost, convexity is a property of set theory. In Machine Learning, the convexity of the loss function refers to the ability of all the golden and predicted labels of the data instances to stay in its epigraph, i.e. above the surface of the function. It is possible if we scale the values of the golden and predicted labels on the loss function graph and treat it as a linear function intersecting with the loss function. That would allow us to choose parameter values along any point between the labels to continue searching for the minimum loss value. If the loss function is convex, then we would step in the desired direction regardless of our choice. Otherwise, the function would have a local maximum on our way (crossing the surface of the linear function), which could lead to raising error if the parameters are chosen in the area of the concave. That could result in the algorithm not converging at a global minimum (or getting into a loop and not converging at all), yielding worse results than theoretically possible.

Another way of learning whether a function is convex is to check if its second derivative is non-negative. According to the lecture notes, the first derivative is the following: $l(p(X)|X) = \sum_{i=1}^{N}[(\sigma(x_i^T\omega) - y_i) \cdot x_i] = 0$. Let's take the second one:

$$\frac{\partial\ l(p(X)|X)}{\partial\ \omega} = \frac{\partial\ \sum_{i=1}^{N}[(\sigma(x_i^T\omega) - y_i)\cdot x_i]}{\partial\omega} = \quad (12)$$

$$= \frac{\partial\ \sum_{i=1}^{N}[x_i\cdot\sigma(x_i^T\omega) - x_i\cdot y_i]}{\partial\omega} = [(x_i\cdot y_i)' = 0] = \sum_{i=1}^{N} x_i\frac{\partial\sigma(x_i^T\omega)}{\partial\omega} = \quad (13)$$

$$= \sum_{i=1}^{N} x_i\cdot\sigma(x_i^T\omega)\cdot(1-\sigma(x_i^T\omega))\cdot x_i^T = \sum_{i=1}^{N} x_i^2\cdot\sigma(x_i^T\omega)\cdot(1-\sigma(x_i^T\omega)) \quad (14)$$

The x_i^2 is certainly non-negative, as well as both σ functions, as its range is within $[0; 1]$. Therefore, the negative log-likelihood function for logistic regression is convex.

2.2 Part (b)

The maximum log-likelihood function looks very similar to the average binary cross-entropy loss. They are the same in the first statement but slightly differ in the second statement. Given that we are currently concerned about a binary task, we could theoretically drop the second statement, as in the two-label setting it is just a compliment of the first one. In that case, they would become equivalent.

3 Task 3

3.1 Part (a)

-

3.2 Part (b)

-