HW1

Problem 1 For linear and logistic regressions, assuming $s_i = \sum_j \beta_j x_{ij} = x_i^\top \beta$.

(1) Find the log-likelihood function

$$J(\beta) = \sum_{i=1}^{n} \log p(y_i \mid s_i = x_i^\top \beta).$$

(2) Find the gradient $J'(\beta)$ by calculating $\partial J/\partial \beta_k$, and write down the gradient ascent algorithm.

Problem 2 For logistic regression based on a simple neural network, with

$$\begin{aligned} s_i &= \beta_0 + \sum_{k=1}^{d} \beta_k h_{ik}, \\ h_{ik} &= \max(0, s_{ik}), \\ s_{ik} &= \alpha_{k0} + \sum_{j=1}^{p} \alpha_{kj} x_{ij}. \end{aligned}$$

(1) Using one-dimensional and two-dimensional input x_i, explain that the above neural network defines a piecewise linear function $s_i = f(x_i)$.

(2) Explain that the last line can be expressed in terms of vectors and matrix.

(3) Let $\theta = (\beta_k, \alpha_{kj}, \forall k, j)$. Find the gradient $J'(\theta)$ by calculating $\partial J/\partial \beta_k$ and $\partial J/\partial \alpha_{kj}$, and write down the gradient ascent algorithm with momentum.

Problem 3 Write Python code to implement the algorithm in Problem 2, where $x_i = (x_{i,1}, x_{i,2}) \sim \text{Uniform}[0,1]^2$, i.e., a unit square, and $y_i = 1(x_{i,1}^2 + x_{i,2}^2 < 1)$, i.e., within a unit circle. Plot the decision boundary defined by the learned neural network.