BIOSTAT 274 Spring 2021 Homework 4

Due 11:59 PM 6/9/2021

Zian ZHUANG

Theoretical Part

Q1.

Consider a neural network for a K class outcome that uses cross-entropy loss. If the network has no hidden layer, show that the model is equivalent to the multinomial logistic model described in ESL Chapter 4.

Answer:

Assume data of n samples $\{(x_i, Y_i)_{i=1,2,3...n}\}$, with $x \in \mathbf{R}^p$ and $Y_i = (Y_{i1}, Y_{i2}, Y_{i3}...Y_{ik})$, $Y_{ik} = 1$ if i-th subject belongs to class k (one-hot encoding tech).

Neural Network

After re-formatting data, we compute the dot product between the vectors containing features and weights, which is called the score (Figure 1).

$$Z_K = \beta_{K0} + \beta_K^T x, \quad K = 1, 2, 3..k$$

Our original weight vector v_K will be an array of 0s given that we do not have any prior information. The weight will be constantly updating when we minimize the cross-entropy loss function.

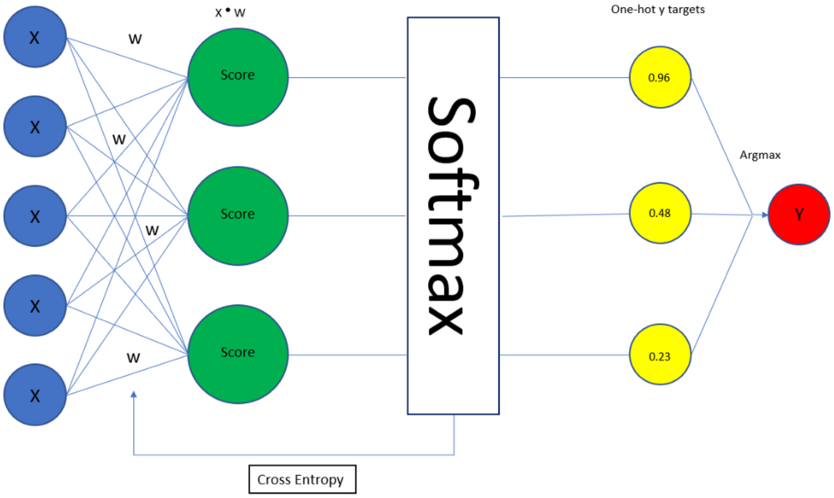

Figure 1: Structure of multinomial logistic regression and network with no hidden layer

Then we implement the Softmax function in order to normalize the scores.

$$\Pr(K|X) = \frac{e^{Z_K}}{\sum_{i=1}^{k} e^{Z_i}}, \quad K = 1, 2, 3...k$$

This exponent normalization function would convert our scores into positives and turn them into probabilities that ranged from 0 to 1. In an array of probability values for each possible result, the argmax of the probability values provides the Y value. For example, in an array of m probabilities, if the j th element has the highest probability for Y_{ik}, then we estimate that j th subject belongs to class k.

Define the $\theta = \{\beta_{K0}, \beta_K\}, K = 1, 2, 3...k$. Then we use the Cross-Entropy is to take the output probabilities (P) and measure the distance from the truth values. Parameters θ are iteratively adjusted accordingly with the aim of minimizing the Cross-Entropy loss. Finally it make the model output be as close as possible to the desired output (truth values). In general, stochastic gradient descent algorithm is applied to the optimization process.

Logistic regression model

The logistic regression model arises from the desire to model the posterior probabilities of the K classes via linear functions in x, while at the same time ensuring that they sum to one and remain in [0, 1]. The model has the form

$$\log \frac{\Pr(K=1|X)}{\Pr(K=k|X)} = \beta_{10} + \beta_1^T x$$

$$\log \frac{\Pr(K=2|X)}{\Pr(K=k|X)} = \beta_{20} + \beta_1^T x$$

$$\log \frac{\Pr(K=3|X)}{\Pr(K=k|X)} = \beta_{30} + \beta_1^T x$$

$$...$$

$$\log \frac{\Pr(K=k-1|X)}{\Pr(K=k|X)} = \beta_{(k-1)0} + \beta_{(k-1)}^T x$$

The model is specified in terms of K-1 log-odds or logit transformations (reflecting the constraint that the probabilities sum to one). Here we uses the last class as the denominator in the odds-ratios, the choice of denominator is arbitrary in that the estimates are equivariant under this choice.

$$e^{\beta_{K0} + \beta_K^T x} = 1, \quad K = k$$

Then we can have,

$$\Pr(K|X) = \frac{e^{\beta_{K0} + \beta_K^T x}}{\sum_{i=1}^{k} e^{\beta_{i0} + \beta_i^T x}}, \quad K = 1, 2, 3, ...k$$

Similarly, we can define the $\theta = \{\beta_{K0}, \beta_K\}, K = 1, 2, 3...k$ and find θ that can minimize the Cross-Entropy loss. Then we find the results of $\{\beta_{K0}, \beta_K\}, K = 1, 2, 3...k$ we got here is equivalent to that calculated from the estimated θ value in the Neural Network. Thus we know that multinomial logistic model is equivalent to the Neural Network that has no hidden layer.