Guidelines for Solving Logistic Regression Problems

Eng. Ahmed Métwalli

February 2025

1 Gradient Descent Method for Logistic Regression

1.1 Step 1: Define the Model

- Logistic regression predicts the probability that $Y = 1$ given the input features.
- The model is defined as:

$$P(Y = 1 \mid X) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

- Where the linear combination is:

$$z = w_0 + w_1 X_1 + w_2 X_2 + \cdots + w_n X_n. \tag{2}$$

- **Explanation of the Linear Combination:**
  The term z is a weighted sum of the input features. Each weight w_j scales its corresponding feature X_j. The bias term w_0 is added to adjust the output independently of the inputs. This linear combination forms the input to the sigmoid function, which then maps z into the range $[0, 1]$ as a probability.

1.2 Step 2: Initialize Parameters

- Set initial weights $w_0, w_1, \ldots, w_n$ (typically to zero).
- Choose a learning rate α.
- Define the number of iterations for the algorithm.

1.3 Step 3: Compute the Loss Function

- Use the Binary Cross-Entropy (Log Loss) function:

$$J(w) = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] \tag{3}$$

- Here, $\hat{y}_i = \sigma(z_i)$ is the predicted probability for the i^{th} example.

1.4 Step 4: Compute the Gradients

- For each parameter w_j, the gradient is computed using the chain rule:

$$\frac{\partial J}{\partial w_j} = \frac{\partial J}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_j}.$$

- We have:
  - **Loss derivative:**

$$\frac{\partial J}{\partial \hat{y}_i} = -\left(\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i} \right).$$

- **Sigmoid derivative:**
  The sigmoid function is given by

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}}.$$

  Its derivative is derived as:

$$\frac{d}{dz_i}\sigma(z_i) = \sigma(z_i)(1 - \sigma(z_i)) = \hat{y}_i(1 - \hat{y}_i).$$

- **Linear combination derivative:**
  Since the linear combination is defined as

$$z_i = w_0 + w_1 X_{i1} + w_2 X_{i2} + \cdots + w_n X_{in},$$

  taking the derivative with respect to w_j gives:

$$\frac{\partial z_i}{\partial w_j} = X_{ij}.$$

  Note that for the bias term w_0, we define $X_{i0} = 1$.

- Multiplying these components:

$$\frac{\partial J}{\partial w_j} = -\left(\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i}\right) \cdot \hat{y}_i(1 - \hat{y}_i) \cdot X_{ij}.$$

- With algebraic manipulation, this expression simplifies to:

$$\frac{\partial J}{\partial w_j} = (\hat{y}_i - y_i)X_{ij}.$$

- For the bias term w_0, the gradient becomes:

$$\frac{\partial J}{\partial w_0} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i).$$

1.5 Step 5: Update Parameters

- Update each weight using the gradient descent rule:

$$w_j = w_j - \alpha\frac{\partial J}{\partial w_j}. \tag{4}$$

1.6 Step 6: Iterate Until Convergence

- Repeat Steps 3–5 for a predefined number of iterations or until the change in parameters is minimal.

2 Iterative Optimization in Logistic Regression

- Unlike linear regression, logistic regression does not have a closed-form solution (i.e., there is no normal equation).
- Iterative methods such as Gradient Descent or Newton-Raphson (IRLS) must be used to find the optimal parameters.

3 Comparison: Gradient Descent vs. Newton-Raphson (IRLS)

Feature	**Gradient Descent**	**Newton-Raphson (IRLS)**
Update Rule	$w_j := w_j - \alpha \frac{\partial J}{\partial w_j}$	$w := w - H^{-1} \nabla J$
Convergence Rate	Slower; requires tuning of α	Faster; uses second-order derivatives
Computation per Iteration	$O(n)$	$O(n^2)$ (due to Hessian matrix computation)
Implementation	Simple	More complex

Table 1: Comparison between Gradient Descent and Newton-Raphson in Logistic Regression