Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India *

2025

Logistic Regression

1 The Logistic Function

The logistic regression model uses the sigmoid function (also called logistic function) to map any real value to the range [0,1]:

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

For a given input feature vector x, the model prediction is:

$$h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \tag{2}$$

where:

- θ is the parameter vector
- x is the input feature vector
- $\theta^T x$ is the dot product $\sum_{i=0}^{n} \theta_i x_i$

This function ensures that predictions remain within the range [0,1], making it suitable for binary classification tasks.

1.1 Visualizing the Sigmoid Function

2 Likelihood Function

For binary classification, we can write the probability of each class as:

$$P(y=1|x;\theta) = h_\theta(x) \tag{3}$$
$$P(y=0|x;\theta) = 1 - h_\theta(x) \tag{4}$$

*Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

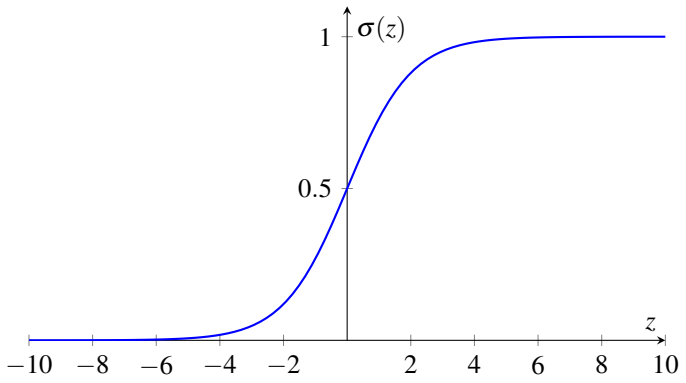

Figure 1: Sigmoid function mapping real values to the range [0,1].

This can be written more compactly as:

$$P(y|x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y} \tag{5}$$

For a training set of m examples, the likelihood function is:

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) = \prod_{i=1}^{m} (h_\theta(x^{(i)}))^{y^{(i)}} (1-h_\theta(x^{(i)}))^{1-y^{(i)}} \tag{6}$$

3 Log-Likelihood and Cost Function

Taking the natural logarithm of the likelihood function (to convert products to sums):

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)})) \right] \tag{7}$$

The cost function (negative log-likelihood) is:

$$J(\theta) = -\frac{1}{m}\ell(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)})) \right] \tag{8}$$

3.1 Illustration of Cost Function

The cost function is designed to penalize incorrect predictions, guiding the optimization process.

4 Gradient Derivation

To find the gradient of the cost function, we need $\frac{\partial}{\partial \theta_j} J(\theta)$.

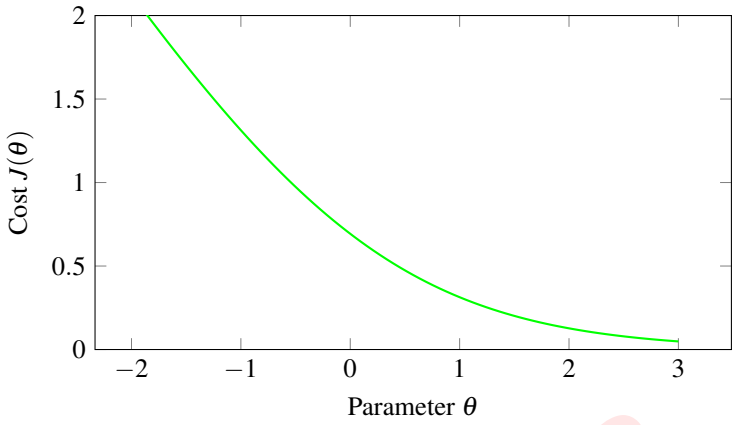

Figure 2: Visualization of the cost function as θ changes.

4.1 Derivative of the Sigmoid Function

$$\frac{d}{dz}\sigma(z) = \sigma(z)(1-\sigma(z)) \tag{9}$$

4.2 Gradient for a Single Example

$$\frac{\partial}{\partial \theta_j}\left[-y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))\right] = (h_\theta(x)-y)x_j \tag{10}$$

4.3 Gradient for the Entire Training Set

$$\frac{\partial}{\partial \theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)} \tag{11}$$

4.4 Detailed Explanation of Chain Rule in Logistic Regression

The chain rule allows us to compute derivatives of composite functions. In logistic regression, the cost function depends on the sigmoid function, which in turn depends on the linear combination $z = \theta^T x$. The process involves three steps:

1. Outer Function (Cost Function):

$$J = -y\log(\sigma(z)) - (1-y)\log(1-\sigma(z)) \tag{12}$$

Differentiating with respect to $\sigma(z)$:

$$\frac{\partial J}{\partial \sigma(z)} = -\frac{y}{\sigma(z)} + \frac{1-y}{1-\sigma(z)} \tag{13}$$

2. Middle Function (Sigmoid Function):

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{14}$$

Differentiating with respect to z:

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z)) \tag{15}$$

3. Inner Function (Linear Combination):

$$z = \theta^T x = \sum_{j=0}^{n} \theta_j x_j \tag{16}$$

Differentiating with respect to θ_j:

$$\frac{\partial z}{\partial \theta_j} = x_j \tag{17}$$

Combining these steps using the chain rule:

$$\frac{\partial J}{\partial \theta_j} = \frac{\partial J}{\partial \sigma(z)} \cdot \frac{\partial \sigma(z)}{\partial z} \cdot \frac{\partial z}{\partial \theta_j} \tag{18}$$

Simplifying:

$$\frac{\partial J}{\partial \theta_j} = (\sigma(z) - y)x_j \tag{19}$$

This systematic approach demonstrates how the chain rule is essential in deriving gradients for logistic regression.

5 Gradient Descent Update Rule

The gradient descent update rule becomes:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{20}$$

where:

- α is the learning rate
- m is the number of training examples

6 Vectorized Form

In vectorized form, the gradient can be written as:

$$\nabla J(\theta) = \frac{1}{m} X^T (h_\theta(X) - y) \tag{21}$$

The vectorized update rule becomes:

$$\theta := \theta - \alpha \frac{1}{m} X^T (h_\theta(X) - y) \tag{22}$$

6.1 Visualization of Gradient Descent

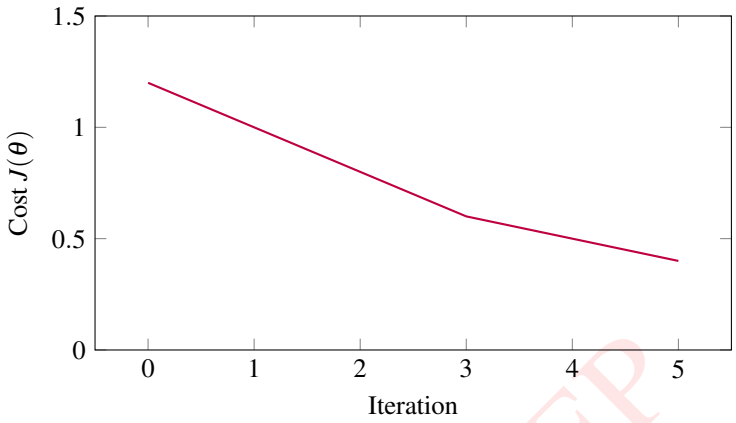

Figure 3: Gradient descent trajectory minimizing the cost function.

7 Frequently Asked Questions (FAQs)

7.1 What is logistic regression used for?

Logistic regression is used for binary classification tasks, where the goal is to predict one of two possible outcomes based on input features.

7.2 How does logistic regression differ from linear regression?

While linear regression predicts continuous values, logistic regression predicts probabilities that are constrained between 0 and 1 using the sigmoid function.

7.3 What is the role of the sigmoid function?

The sigmoid function maps any real-valued input into the range [0,1], allowing logistic regression to model probabilities.

7.4 Why do we use the log-likelihood instead of the likelihood?

The log-likelihood simplifies computations by converting products into sums, making optimization more manageable.

7.5 What is the cost function in logistic regression?

The cost function is the negative log-likelihood, which measures how well the model predicts the training data. It is minimized during training.

7.6 What are common optimization methods for logistic regression?

Gradient descent and its variants (e.g., stochastic gradient descent) are commonly used to optimize the parameters of logistic regression.

7.7 How can overfitting be avoided in logistic regression?

Regularization techniques, such as L1 (lasso) and L2 (ridge), are used to prevent overfitting by penalizing large parameter values.