Artificial Intelligence and Automation

Understanding Logistic Regression

Ph.D. Gerardo Marx Chávez-Campos

Instituto Tecnológico de Morelia: Ing. Mecatrónica

Mathematical Formulation

Logistic Regression is a classification algorithm used to predict binary outcomes (0 or 1). Unlike linear regression, which predicts continuous values, logistic regression applies a sigmoid function to model probabilities:

$$h_\theta(X) = \frac{1}{1 + e^{-X\theta}} \tag{1}$$

Where:

- $h_\theta(X)$ is the probability of class $y = 1$ given input X.
- θ are the model parameters.
- X is the feature vector.

Understanding the Likelihood Function I

In logistic regression, we model the probability that $y = 1$ given x as:

$$P(y = 1|x; \theta) = h_\theta(x) = \frac{1}{1 + e^{-X\theta}} \quad (2)$$

Similarly, the probability that $y = 0$ is:

$$P(y = 0|x; \theta) = 1 - h_\theta(x) \quad (3)$$

Understanding the Likelihood Function II

Thus, for a single training example (x^i, y^i), we can write:

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

This formula works because:

- If $y^i = 1$, then $P(y^i|x^i;\theta) = h_\theta(x^i)$.
- If $y^i = 0$, then $P(y^i|x^i;\theta) = 1 - h_\theta(x^i)$.

Understanding the Likelihood Function III

For the entire dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, assuming independence of training examples, the likelihood function (joint probability of all data points) is:

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) \tag{5}$$

Expanding this:

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

Log-Likelihood and Cost Function

Since products can be numerically unstable (due to very small probabilities), we take the log of the likelihood function to obtain the log-likelihood:

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

MLE aims to maximize the log-likelihood $\ell(\theta)$. Instead of maximizing it, we minimize the negative log-likelihood, which is called the cost function:

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

Gradient Descent

To minimize $J(\theta)$, we compute its gradient[1]:

$$\frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{9}$$

Gradient descent update rule:

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_j} \tag{10}$$

where α is the learning rate.

[1] Procedure in another beamer

Compute the Partial Derivative

We differentiate the cost function with respect to θ_j:

$$\frac{\partial J}{\partial \theta_j} = -\frac{1}{m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta_j} \left[y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right) \right] \tag{11}$$

Differentiate the Log Terms

Using the chain rule:

1. Derivative of $\log h_\theta(x)$:

$$\frac{\partial}{\partial \theta_j} \log h_\theta(x^{(i)}) = \frac{1}{h_\theta(x)} \cdot \frac{\partial h_\theta(x)}{\partial \theta_j} \tag{12}$$

2. Derivative of $\log(1 - h_\theta(x))$:

$$\frac{\partial}{\partial \theta_j} \log(1 - h_\theta(x)) = \frac{-1}{1 - h_\theta(x)} \cdot \frac{\partial h_\theta(x)}{\partial \theta_j} \tag{13}$$

Compute the Derivative of Sigmoid Function

The sigmoid function is:

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

Differentiating $h_\theta(x)$:

$$\frac{d}{dz}\sigma(z) = h_\theta(z)(1 - h_\theta(z))\frac{d}{dz}z \tag{15}$$

Thus,

$$\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} = h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))x_j^{(i)} \tag{16}$$

Substitute Back into the Gradient

Now, substituting back:

$$\frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \tag{17}$$

Final Gradient Formula

The final gradient formula in vector form is:

$$\nabla J(\theta) = \frac{1}{m} X^T (h_\theta(X) - y) \tag{18}$$

where:

- X is the feature matrix (each row is an input sample).
- $h_\theta(X)$ is the vector of predictions.
- y is the vector of true labels.

This formula tells us how to update the parameters:

$$\theta := \theta - \alpha \nabla J(\theta) \tag{19}$$

where α is the learning rate.

Conclusion

- We started with the logistic regression cost function.
- We computed its derivative using the chain rule and the sigmoid derivative.
- The resulting gradient formula looks similar to linear regression but applies to logistic regression probabilities.
- This formula is used in gradient descent to optimize θ.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN: 9780387310732. URL: https://www.microsoft.com/en-us/research/people/cmbishop/.

[2] David R. Cox. "The Regression Analysis of Binary Sequences". In: *Journal of the Royal Statistical Society: Series B* 20.2 (1958), pp. 215–242.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. Springer, 2009. URL: https://web.stanford.edu/~hastie/ElemStatLearn/.

[4] Andrew Ng. *Machine Learning Course*. Coursera. 2011. URL: https://www.coursera.org/learn/machine-learning.

[5] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd. Springer, 2006. ISBN: 9780387303031.