Logistic Regression Implementation in C

Madhav Kataria

March 18, 2024

1 Introduction

1.1 Logistic Regression Overview

Logistic regression is a classification algorithm used for predicting the probability of a categorical dependent variable belonging to a certain class. Unlike linear regression, it uses the logit or sigmoid function to map linear model outputs to probabilities between 0 and 1.

1.2 Applications

Typical use cases for logistic regression include predicting customer churn, credit risk assessment, disease diagnosis, and spam detection.

2 Code Purpose

2.1 Objective

The provided code implements a basic logistic regression model from scratch in the C programming language. It demonstrates core concepts and calculations involved in this machine learning technique.

2.2 Data Assumption

The code assumes a "data.csv" file containing training data with features and target labels.

3 Code Analysis

3.1 Structures

- `LR` struct: The core structure representing the logistic regression model. It stores:
  - `N`: Number of training samples

- `n_in`: Number of input features
- `n_out`: Number of output classes
- `W`: Weight matrix (coefficients)
- `b`: Bias vector

3.2 Functions

- `LR_construct`: Initializes the `LR` structure, allocating memory for weights and biases.
- `LR_destruct`: Frees the allocated memory.
- `LR_train`: Implements the training process of the model using gradient descent. This includes calculation of predicted probabilities, softmax function for multi-class probability normalization, gradient computation, and updating weights and biases.
- `LR_softmax`: Normalizes probabilities to sum to 1 ensuring they are valid probability distributions.
- `LR_predict`: Uses the trained model to generate predictions for new input samples.
- `test_lr`: Encapsulates the main working logic of the code, including reading data from "data.csv", constructing the LR model, iterative training process over multiple epochs, and evaluation.

3.3 Logistic Regression Formulas

- Linear Model: $z = b + w_1 * x_1 + w_2 * x_2 + ... + w_n * x_n$, where z is the linear combination of input features, w_i are weights, and b is the bias.
- Sigmoid (Logistic) Function: $p = \frac{1}{1+\exp(-z)}$, where p represents the predicted probability.
- Loss Function: Typically cross-entropy loss is used for logistic regression.
- Gradient Descent Updates: Formulas for updating weights and biases during the training process.

4 Conclusion

In conclusion, the provided code presents an implementation of logistic regression in C. While it can work with most datasets, there are areas that can be optimized for improved performance.