Introduction to Logistic Regression for Binary Classification

1 Introduction

Unlike its name suggests logistic regression is a statistical method used for classification problems (not regression problems), where the goal is to predict one of many possible outcomes. In this document we will be focusing on the simplest case where the goal is to predict one of two possible outcomes, this is called binary classification. This problem is common in fields such as medicine where we might need to predict whether a patient does or does not have a specific illness given some observed data.

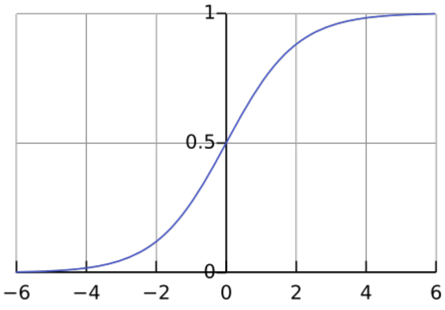
This document provides an overview of logistic regression for binary classification, including prediction, likelihood, gradient derivation, optimization using gradient descent, and evaluation using performance metrics.

2 Prediction with Logistic Regression

In binary classification, logistic regression aims to model the probability that an input $\mathbf{x} \in \mathbb{R}^D$ belongs to class 1. Let the output be denoted as $y \in \{0, 1\}$. The logistic regression model is given by the following equation:

$$P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}),$$

where: - $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid function. - $\mathbf{w} \in \mathbb{R}^D$ is the weight vector (parameters) of the model. - $\mathbf{x} \in \mathbb{R}^D$ is the feature vector, with 1 already appended to the beginning of the vector so that we can model the bias term.



The probability that the outcome is $y = 0$ is simply:

$$P(y = 0|\mathbf{x}) = 1 - \sigma(\mathbf{w}^T\mathbf{x}).$$

3 Likelihood of Logistic Regression

To estimate the parameters $\mathbf{w}$ of the logistic regression model, we use the likelihood function. Given a dataset with N training examples, $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the likelihood of the data under the model is:

$$L(\mathbf{w}) = \prod_{i=1}^{N} P(y_i|\mathbf{x}_i) = \prod_{i=1}^{N} \sigma(\mathbf{w}^T\mathbf{x}_i)^{y_i} \left(1 - \sigma(\mathbf{w}^T\mathbf{x}_i)\right)^{1-y_i}.$$

The log-likelihood function is often easier to work with and more numerically stable, this is given by:

$$\log L(\mathbf{w}) = \sum_{i=1}^{N} \left[y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \log \left(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \right].$$

Maximizing the log-likelihood is equivalent to minimizing the negative log-likelihood, which is the objective of logistic regression.

4 Derivation of the Gradient of the Negative Log-Likelihood

The negative log-likelihood function is:

$$-\log L(\mathbf{w}) = -\sum_{i=1}^{N} \left[y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \log \left(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \right].$$

To minimize this, we need to compute the gradient of the negative log-likelihood with respect to $\mathbf{w}$. The gradient of the negative log-likelihood is:

$$\nabla_{\mathbf{w}}(-\log L(\mathbf{w})) = -\sum_{i=1}^{N} \left[y_i \frac{\partial}{\partial \mathbf{w}} \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \frac{\partial}{\partial \mathbf{w}} \log \left(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \right].$$

First, recall that the derivative of the sigmoid function is:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

Using this, we can compute the gradients for each term:

$$\frac{\partial}{\partial \mathbf{w}} \log \sigma(\mathbf{w}^T \mathbf{x}_i) = \frac{\sigma'(\mathbf{w}^T \mathbf{x}_i)\mathbf{x}_i}{\sigma(\mathbf{w}^T \mathbf{x}_i)} = \left(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \mathbf{x}_i,$$

and

$$\frac{\partial}{\partial \mathbf{w}} \log \left(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) = -\frac{\sigma'(\mathbf{w}^T \mathbf{x}_i)\mathbf{x}_i}{1 - \sigma(\mathbf{w}^T \mathbf{x}_i)} = -\sigma(\mathbf{w}^T \mathbf{x}_i)\mathbf{x}_i.$$

Thus, the gradient of the negative log-likelihood becomes:

$$\nabla_{\mathbf{w}}(-\log L(\mathbf{w})) = \sum_{i=1}^{N} \left(\sigma(\mathbf{w}^T \mathbf{x}_i) - y_i\right) \mathbf{x}_i.$$

5 Optimization with Gradient Descent

To optimize the parameters $\mathbf{w}$, we can use gradient descent. The update rule for gradient descent is:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}}(-\log L(\mathbf{w}_t)),$$

where η is the learning rate and t is the iteration index.

Substituting the expression for the gradient:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \sum_{i=1}^{N} \left(\sigma(\mathbf{w}_t^T \mathbf{x}_i) - y_i\right) \mathbf{x}_i.$$

This process is repeated until convergence, where the change in the weights becomes small or the maximum number of iterations is reached.

6 Performance Metrics

To evaluate the performance of a binary classifier, several metrics can be used. These include:

6.1 Confusion Matrix

The confusion matrix summarizes the predictions of a binary classifier:

	Predicted: 0	Predicted: 1
Actual: 0	True Negative (TN)	False Positive (FP)
Actual: 1	False Negative (FN)	True Positive (TP)

From the confusion matrix, the following metrics can be derived:

6.2 False Positive Rate (FPR)

The false positive rate is the proportion of actual negatives that are incorrectly classified as positives:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

6.3 True Positive Rate (TPR) or Recall

The true positive rate (also known as recall or sensitivity) is the proportion of actual positives that are correctly classified:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

6.4 False Negative Rate (FNR)

The false negative rate is the proportion of actual positives that are incorrectly classified as negatives:

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}.$$

6.5 True Negative Rate (TNR)

The true negative rate (also known as specificity) is the proportion of actual negatives that are correctly classified:

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}.$$

6.6 Precision

Precision is the proportion of predicted positives that are actually positive:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$

6.7 Accuracy

Accuracy is the overall proportion of correct predictions:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}.$$