

Lecture 3c: Logistic Regression for Binary Classification Problems

*Lecturer: Jeffrey Varner***Disclaimer:** *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

1 Introduction

In this lecture, we will introduce logistic regression for binary classification problems. We will start by discussing the logistic regression model and then the maximum likelihood estimation of the model parameters. We will also discuss the gradient descent algorithm for estimating the model parameters. Finally, we will discuss evaluating the logistic regression model using the confusion matrix, accuracy, precision, recall, and the F1 score. The key concepts covered in this lecture include:

- **Logistic regression** is a statistical method used for binary classification that models the relationship between a dependent categorical variable (label) and one or more independent variables (features) by estimating probabilities through the logistic function.
- **Maximum likelihood estimation (MLE)** is a statistical technique to estimate the parameters of a probability distribution by maximizing the likelihood function, thereby determining the parameter values that make the observed data most probable.
- **Gradient descent** is an optimization algorithm used to minimize a function by iteratively adjusting parameters in the opposite direction of the gradient. Iteration continues until a local minimum of the function is found.
- **Performance assesment.** Evaluating the performance of a logistic regression model involves assessing its accuracy and predictive power through various metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve. These metrics provide insights into how well the model distinguishes between the binary classes and help fine-tune its parameters for improved performance.

2 Logistic Regression

Logistic regression is a statistical method used for binary classification problems, where the dependent variable (label) is a binary categorical variable (e.g., ± 1 , etc), and the independent variables (features) are continuous or categorical variables. Unlike the Perceptron model, which outputs the class label directly, logistic regression models the probability that a given input belongs to a particular class based on the input features. The logistic regression model estimates the probability that a given input belongs to a particular class based on the input features. In particular, logistic regression is a discriminative model, which means it directly models the conditional probability of the label given the features, i.e., $P(y|\mathbf{x})$. This is in contrast to generative models, e.g., Naive Bayes, which we'll explore later, which model the joint probability of the features and the label, i.e., $P(y, \mathbf{x}) = P(\mathbf{x}|y) \cdot P(y)$. The logistic regression model uses the logistic function to model the probability of the binary label $y \in \{-1, +1\}$ given the feature vector \mathbf{x} :

$$P(y|\mathbf{x}; \theta) = \frac{1}{1 + e^{-y \cdot \theta^T \mathbf{x}}} \quad (1)$$

where $\theta \in \mathbb{R}^n$ is an (unknown) parameter vector (that we need to estimate somehow), and e is the base of the natural logarithm. The logistic function is a sigmoid function that maps the input, i.e., $-y \cdot \theta^T \mathbf{x}$ to the range $[0, 1]$, which is suitable for modeling probabilities. The logistic regression model predicts the label $y \in \{-1, +1\}$ for a given feature vector \mathbf{x} by comparing the probability $P(y|\mathbf{x}; \theta)$ to a threshold, e.g., 0.5. The model predicts the positive class if the probability exceeds the threshold ($y = 1$). Otherwise, it predicts the negative class ($y = -1$). The logistic regression model is trained by estimating the parameters θ that maximize the likelihood of the observed labels given the features. The next section will discuss the maximum likelihood estimation of the logistic regression model parameters.

2.1 Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation (MLE) is a technique to estimate the parameters of a probability distribution by maximizing the likelihood function. In logistic regression, MLE estimates the parameters of the logistic regression model that make the observed label conditioned on the features the most probable. Given a set of training examples $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ is an n -dimensional feature vector and $y_i \in \mathbb{R}$ is the binary (scalar) label, the likelihood function is defined as:

$$\mathcal{L}(\theta) = \prod_{i=1}^m P(y_i|\mathbf{x}_i; \theta) \quad (2)$$

where $P(y_i|\mathbf{x}_i; \theta)$ is the probability of observing the label y_i given the feature vector \mathbf{x}_i and the model parameters θ . It's hard to maximize the likelihood function directly (because of the product), so we take the logarithm of the likelihood function to simplify the optimization:

$$\log \mathcal{L}(\theta) = \sum_{i=1}^m \log P(y_i|\mathbf{x}_i; \theta) \quad (3)$$

The \log is a monotonic function, so maximizing the log-likelihood is equivalent to maximizing the likelihood. The logistic regression model uses the logistic function to model the probability of the binary label:

$$P(y_i|\mathbf{x}_i; \theta) = \frac{1}{1 + e^{-y_i \cdot (\theta^T \mathbf{x}_i)}} \quad (4)$$

where $\theta \in \mathbb{R}^n$ is the parameter vector, θ^T is the transpose of θ , and e is the base of the natural logarithm. Substituting the $P(y_i|\mathbf{x}_i; \theta)$ function into the log-likelihood function gives:

$$\log \mathcal{L}(\theta) = - \sum_{i=1}^m \log \left(1 + e^{-y_i \cdot \theta^T \mathbf{x}_i} \right) \quad (5)$$

where we inverted the $P(y_i|\mathbf{x}_i; \theta)$ function to get the log-likelihood function. The maximum likelihood estimation (MLE) of the logistic regression model parameters θ^* is obtained by maximizing the log-likelihood function $\log \mathcal{L}(\theta)$:

$$\theta^* = \arg \max_{\theta} \left[- \sum_{i=1}^m \log \left(1 + e^{-y_i \cdot \theta^T \mathbf{x}_i} \right) \right] \quad (6)$$

There is no closed-form analytical solution to this optimization problem, so we must use a numerical algorithm to estimate the model parameters. To solve this optimization problem, we can use the gradient descent algorithm (or one of many other approaches) to iteratively update the parameters θ to maximize the log-likelihood function, or alternatively, minimize the negative log-likelihood function, i.e., $l(\theta) = \sum_{i=1}^m \log \left(1 + e^{-y_i \cdot \theta^T \mathbf{x}_i} \right)$.

We'll start by discussing the gradient descent algorithm and then consider some alternatives to gradient descent.

2.2 Gradient Descent

Gradient descent is an optimization algorithm that minimizes a function by iteratively adjusting the parameters in the opposite direction of the gradient. Suppose there exists an objective function $l(\theta)$ that we want to minimize with respect to the parameter θ , i.e., the negative log-likelihood function. In general, an objective function measures the difference between the predicted values and the observed values in some way, e.g., the mean squared error (MSE), the cross-entropy loss, or the negative log-likelihood. In logistic regression, the objective function is the negative log-likelihood function, which measures the difference between the predicted probabilities and the observed labels. However, whatever form the objective function takes, we assume that it is differentiable and that we can compute the gradient, i.e., $\nabla l(\theta)$ for the negative log-likelihood function, which points in the direction of the steepest increase of the function. This gives us a way to update the parameters to minimize the objective function using the update rule:

$$\theta_{k+1} = \theta_k - \alpha(k) \cdot \nabla l(\theta_k) \quad \text{where } k = 0, 1, 2, \dots$$

The (hyper) parameter $\alpha(k) > 0$ is the learning rate (which can be a function of the iteration count k), and $\nabla l(\theta)$ is the gradient of the negative log-likelihood function with respect to the parameters. We iterate until a stopping criterion is met, i.e., $\|\theta_{k+1} - \theta_k\| \leq \epsilon$, the maximum number of iterations is reached, or some other stopping criterion is met.

2.3 Alternatives to Gradient Descent

The central issue with gradient descent is that it can be slow to converge, especially when the objective function is non-convex or has many local minima. The choice of the learning rate α is crucial, as a too-large value can cause the algorithm to diverge, while a too-small value can slow down convergence. Further, the objective function may not be differentiable, or the gradient may be challenging to compute. In these cases, alternative heuristic optimization algorithms can be used to estimate the model parameters. Let's walk through some of the alternatives to gradient descent.

Simulated Annealing

Simulated annealing is a probabilistic optimization algorithm inspired by the physical process of heating and then slowly cooling materials to minimize defects, allowing for a thorough exploration of the solution space by accepting worse solutions with a certain probability of avoiding local minima. This method effectively solves complex optimization problems with large search spaces, where traditional techniques may struggle to find the global optimum.

Genetic Algorithms

Genetic algorithms (GAs) are adaptive heuristic search techniques inspired by natural selection and genetics principles. They are designed to solve optimization and search problems by iteratively evolving a population of candidate solutions through selection, crossover, and mutation. By mimicking the evolutionary process, GAs aim to improve solution quality over generations, making them particularly effective for complex problems that may be discontinuous, non-differentiable, or highly nonlinear.

Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a meta-heuristic optimization algorithm inspired by the social behavior of birds and fish, which utilizes a population of candidate solutions, referred to as particles, that move through the search space to find optimal solutions. Each particle adjusts its position based on its own experience and the collective knowledge of the swarm, allowing for efficient exploration and exploitation of the solution space to address complex optimization problems across various fields.

3 Performance Evaluation

Evaluating the performance of a logistic regression model involves assessing its accuracy and predictive power through various metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve. The accuracy of a model is the proportion of correctly classified instances, while precision measures the proportion of true positive predictions among all positive predictions. The recall is the proportion of true positive predictions among all actual positive instances, and the F1-score is the harmonic mean of precision and recall. The area under the ROC curve measures the model's ability to distinguish between the two classes, with a higher AUC indicating better performance.

4 Summary and Conclusions

In this lecture, we introduced logistic regression for binary classification problems and discussed the maximum likelihood estimation of the model parameters. We also covered the gradient descent algorithm for estimating model parameters and evaluated the logistic regression model using the confusion matrix, accuracy, precision, recall, and F1 score. Gradient descent is an optimization algorithm employed to minimize a function by iteratively adjusting parameters in the opposite direction of the gradient. However, various other optimization algorithms can estimate model parameters without relying on the gradient. For instance, simulated annealing, genetic algorithms, and particle swarm optimization are all methods that can be utilized to estimate the model parameters. Finally, we examined the logistic regression model's performance using various metrics, including accuracy, precision, recall, F1 score, and the area under the ROC curve. Logistic regression is a powerful tool for binary classification and is widely applied in numerous fields, including healthcare, finance, and marketing.