Machine Learning

Dr. Indu Joshi

Assistant Professor at
Indian Institute of Technology Mandi

4 March 2025

Classification

- The classification problem is just like the regression problem, except that the values y we now want to predict take on only a small number of discrete values.
- For now, we will focus on the *binary classification problem* in which y can take on only two values, 0 and 1.
- For instance, if we are trying to build a spam classifier for email, then $x^{(i)}$ may be some features of a piece of email, and y may be 1 if it is a piece of spam mail, and 0 otherwise.
- 0 is also called the *negative class*, and 1 the *positive class*, and they are sometimes also denoted by the symbols "-" and "+."
- Given $x^{(i)}$, the corresponding $y^{(i)}$ is also called the *label* for the training example.

Logistic Regression

- We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given x.
- However, intuitively, it doesn't make sense for $h_\theta(x)$ to take values larger than 1 or smaller than 0 when we know that $y \in \{0, 1\}$.
- To fix this, let's change the form for our hypotheses $h_\theta(x)$. We will choose

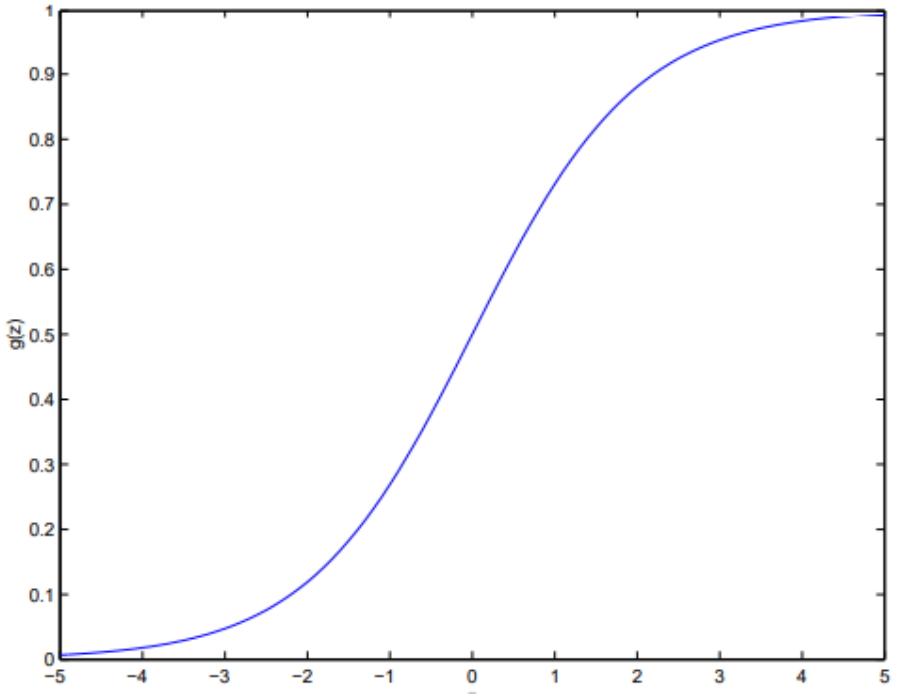
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

Logistic Regression

where

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

is called the *logistic function* or the *sigmoid function*.



Logistic Regression

- Notice that $g(z)$ tends towards 1 as $z \to \infty$, and $g(z)$ tends towards 0 as $z \to -\infty$.
- Moreover, $g(z)$, and hence also $h(x)$, is always bounded between 0 and 1. As before, we are keeping the convention of letting $x_0 = 1$, so that

$$\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j.$$

Logistic Regression

- Before moving on, here's a useful property of the derivative of the sigmoid function, which we write as $g'(z)$:

$$g'(z) = \frac{d}{dz}\frac{1}{1+e^{-z}} = \frac{1}{(1+e^{-z})^2} \cdot (e^{-z})$$

$$= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}}\right) = g(z)(1-g(z)).$$

- So, given the logistic regression model, how do we fit θ for it?
- Following how we saw least squares regression could be derived as the maximum likelihood estimator under a set of assumptions, let's endow our classification model with a set of probabilistic assumptions, and then fit the parameters via maximum likelihood.

Logistic Regression

- Linear regression is not ideal for classification as it outputs continuous values.
- Logistic Regression models the probability of class membership.
- Uses the sigmoid function to map outputs between 0 and 1:

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

- This ensures output values represent probabilities.

Mathematical Formulation

- Given input features $\mathbf{x}$ and output y, logistic regression models:

$$h_\theta(x) = P(y = 1 \mid \mathbf{x}) = \sigma(\theta^T \mathbf{x})$$

- The function $\sigma(z)$ is called the sigmoid function:

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

- The output is interpreted as a probability, with values between 0 and 1.

Decision Boundary and Interpretation

- The decision boundary is where $P(y = 1 \mid \mathbf{x}) = 0.5$, which simplifies to:

$$\theta_0 + \sum_{j=1}^{n} \theta_j x_j = 0$$

- This represents a linear decision boundary.

Maximum Likelihood Estimation for Logistic Regression

Let us assume that:

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

$$\implies p(y|x; \theta) = (h_\theta(x))^y \, (1 - h_\theta(x))^{1-y}$$

Assuming that the m training examples were generated independently, we can write the likelihood of the parameters as:

$$\begin{aligned} L(\theta) &= p(\mathbf{y}|X; \theta) \\ &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta) \\ &= \prod_{i=1}^{m} \left(h_\theta(x^{(i)}) \right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)}) \right)^{1-y^{(i)}} \end{aligned}$$

Maximum Likelihood Estimation for Logistic Regression

To simplify optimization, we take the log-likelihood:

$$\begin{aligned}\ell(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m} \left[y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \right]\end{aligned}$$

To maximize the likelihood, we use **gradient ascent**. The update rule is:

$$\theta := \theta + \alpha \nabla_\theta \ell(\theta)$$

where α is the learning rate. Note the positive sign, since we are maximizing the function.

Maximum Likelihood Estimation for Logistic Regression

Starting with a single training example (x, y), we compute the derivative:

$$\begin{aligned}
\frac{\partial}{\partial \theta_j}\ell(\theta) &= \left(\frac{y}{g(\theta^T x)} - \frac{(1-y)}{1-g(\theta^T x)}\right)\frac{\partial}{\partial \theta_j} g(\theta^T x) \\
&= \left(\frac{y}{g(\theta^T x)} - \frac{(1-y)}{1-g(\theta^T x)}\right) g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial \theta_j}\theta^T x \\
&= \Big(y(1-g(\theta^T x)) - (1-y)g(\theta^T x)\Big)\, x_j \\
&= (y - h_\theta(x))x_j
\end{aligned}$$

This leads to the stochastic gradient ascent update rule:

$$\theta_j := \theta_j + \alpha(y - h_\theta(x))x_j$$

Stochastic Gradient Ascent Rule

The stochastic gradient ascent rule:

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}$$

- Comparing this to the LMS update rule, we observe that it appears identical.
- However, this is not the same algorithm, because $h_\theta(x^{(i)})$ is now a non-linear function of $\theta^T x^{(i)}$.

Advantages of Logistic Regression

- Simple and interpretable model.
- Works well with small datasets.
- Outputs probabilities directly, enabling better decision-making.

Limitations of Logistic Regression

- Assumes a linear decision boundary, limiting its flexibility.

Summary

- Logistic Regression is widely used for binary classification.
- Uses the sigmoid function to model probability.
- Trained using maximum likelihood estimation and gradient ascent.
- Performance evaluated using metrics like AUC-ROC and F1-score.

Thank You

Contact: indujoshi@iitmandi.ac.in