Machine Learning Using Python

Lesson 4: Logistic Regression and Optimal Decisions

Marcel Scharth

The University of Sydney Business School

Lesson 4: Logistic Regression and Optimal Decisions

1. Optimal decisions

2. Logistic regression

Optimal decisions

Loss matrix (key concept)

In most business problems, we will want to specify a **loss matrix** or **cost-benefit matrix** for classification as follows.

		Classification	
		$\widehat{Y} = 0$	$\widehat{Y} = 1$
Actual	$Y = 0$	L_{TN}	L_{FP}
	$Y = 1$	L_{FN}	L_{TP}

Example: credit scoring

In credit scoring, we want to classify a loan applicant as creditworthy ($Y = 1$) or not ($Y = 0$) based on the probability that the customer will not default.

		Classification	
		$\widehat{Y} = 0$	$\widehat{Y} = 1$
Actual	$Y = 0$	Default loss avoided	Default loss
	$Y = 1$	Profit opportunity lost	Profit

A false positive is a more costly error than a false negative for this business scenario. Our decision making should therefore take this into account.

Binary Classifier

$$\widehat{y}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } P(Y=1|X=\boldsymbol{x}) > \tau, \\ 0 \text{ if } P(Y=1|X=\boldsymbol{x}) \leq \tau. \end{cases}$$

Our classifier is a function of τ. We therefore should carefully the optimal τ according to our loss matrix

Optimal decision

Let $\pi = P(Y = 1|X = \boldsymbol{x})$ to simplify the notation. We compare the expected loss from each decision as function of τ,

$$E\left[L(Y, \delta_\tau(\boldsymbol{x}))|X = \boldsymbol{x}\right] = \begin{cases} \pi L_{\mathsf{TP}} + (1-\pi)L_{\mathsf{FP}} \text{ if } \widehat{Y}_\tau(\boldsymbol{x}) = 1, \\ \pi L_{\mathsf{FN}} + (1-\pi)L_{\mathsf{TN}} \text{ if } \widehat{Y}_\tau(\boldsymbol{x}) = 0. \end{cases}$$

Optimal decision (key concept)

The optimal decision threshold corresponds to the probability value π such that the loss from a positive or negative classification is equal.

$$\tau^* L_{\mathsf{TP}} + (1 - \tau^*) L_{\mathsf{FP}} = \tau^* L_{\mathsf{FN}} + (1 - \tau^*) L_{\mathsf{TN}}$$

Solving the equation, the optimal threshold is

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{(L_{\mathsf{FP}} - L_{\mathsf{TN}}) + (L_{\mathsf{FN}} - L_{\mathsf{TP}})}$$

Logistic regression

Regression models for classification

Suppose that we want the specify a discriminative model for binary classification. The response Y conditionally follows the Bernoulli distribution,

$$Y = \begin{cases} 1 \text{ with probability } P(Y = 1|X = \boldsymbol{x}) \\ 0 \text{ with probability } 1 - P(Y = 1|X = \boldsymbol{x}) \end{cases}$$

How to model the conditional probability $P(Y = 1|X = \boldsymbol{x})$ as a function of the predictors?

Regression models for classification

One option is to specify a linear regression model

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon.$$

This is called the **linear probability model**. However, there are a few reasons why we want to move beyond this framework.

Why not the linear probability model?

1. There is no guarantee that a linear probability model will generate probabilities between zero and one, since the regression function $\beta_0 + \sum_{j=1}^{p} \beta_j x_j$ is unconstrained. In other words, the linearity assumption does not hold.

2. If we know that the response conditionally follows a Bernoulli distribution, it is statistically inefficient to ignore this information and fit the model by OLS.

3. The linear probability approach does not generalise to categorical responses with more than two classes.

Logistic regression (key concept)

The **logistic regression model** is

$$Y|X = \boldsymbol{x} \sim \text{Ber}(p(\boldsymbol{x})),$$

where

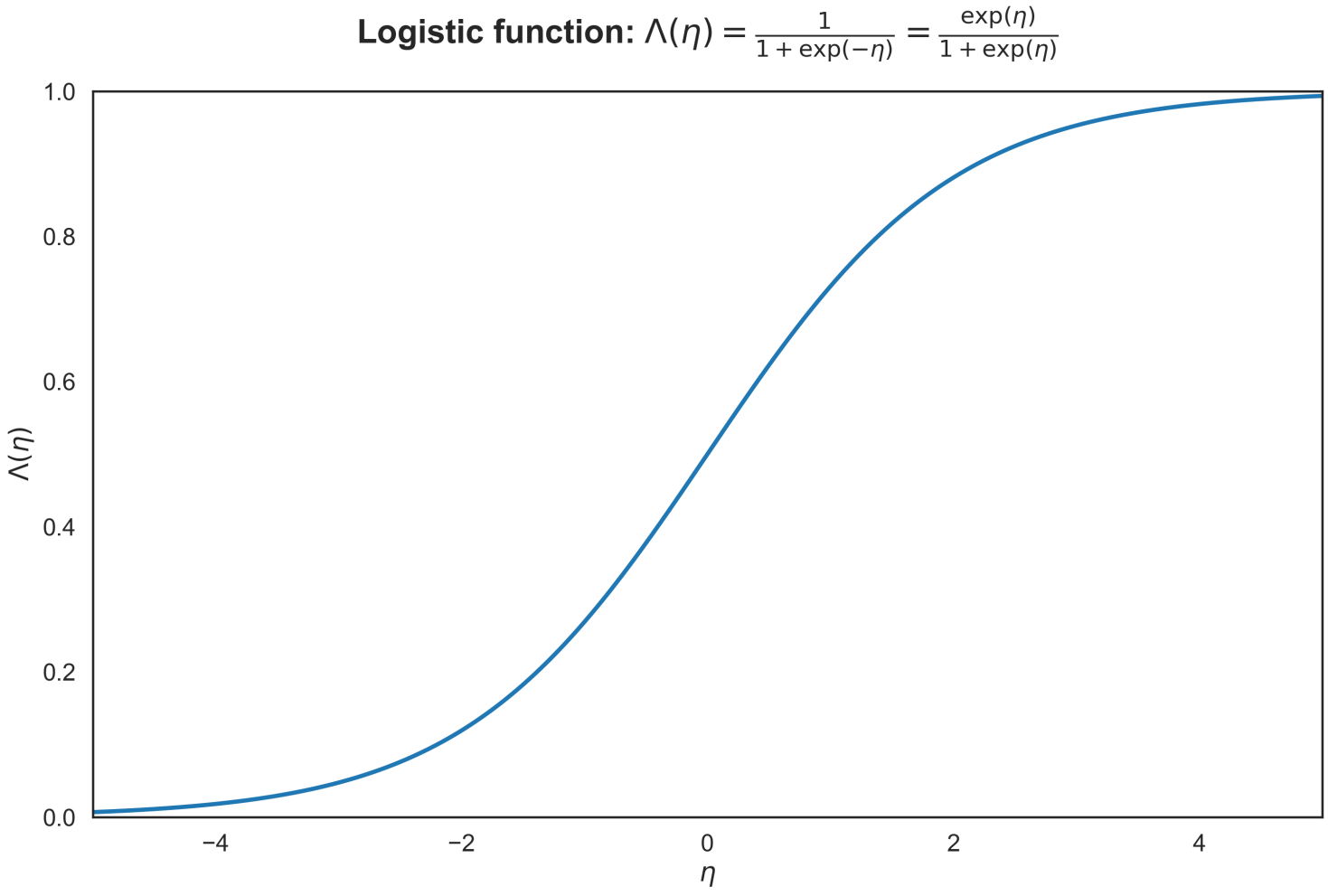
$$p(\boldsymbol{x}) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}.$$

The **logistic function**

$$\frac{\exp(a)}{1 + \exp(a)} = \frac{1}{1 + \exp(-a)}$$

constrains the probability to be between zero and one.

Logistic function



Logistic Regression

Define the **odds ratio** as

$$\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}.$$

We can show that

$$\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})} = \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right).$$

The logistic regression model therefore specifies a linear model for the log odds,

$$\log\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j,$$

where we call the left-hand side the logit transformation of the probability.

Maximum likelihood estimation (technical)

We estimate the logistic regression model by maximum likelihood. A Bernoulli random variable Y has probability mass function

$$p(y; \pi) = \pi^y (1 - \pi)^{1-y}.$$

In the context of the logistic regression model, the probability mass function for a training case i is therefore

$$p(y_i | x_i) = p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i}.$$

Maximum likelihood estimation

The likelihood function is

$$\begin{aligned}
\ell(\boldsymbol{\beta}) &= p(y_1|\boldsymbol{x}_1)\, p(y_2|\boldsymbol{x}_2)\, \ldots\, p(y_N|\boldsymbol{x}_N) \\
&= \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i) \\
&= \prod_{i=1}^{N} p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i} \\
&= \prod_{i=1}^{N} \left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)} \right)^{y_i} (1 - p(\boldsymbol{x}_i))
\end{aligned}$$

Maximum likelihood estimation

The log-likelihood is

$$
\begin{aligned}
L(\boldsymbol{\beta}) &= \log \left(\prod_{i=1}^{N} \left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)} \right)^{y_i} (1 - p(\boldsymbol{x}_i)) \right) \\
&= \sum_{i=1}^{N} y_i \log \left(\frac{p(\boldsymbol{x}_i)}{1 - p(\boldsymbol{x}_i)} \right) + \log \left(1 - p(\boldsymbol{x}_i)\right) \\
&= \sum_{i=1}^{N} y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) - \log \left(1 + \exp \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right)
\end{aligned}
$$

The negative log-likelihood $-L(\boldsymbol{\beta})$ is known as the **cross-entropy loss function** or log-loss in machine learning.

Regularised logistic regression

Regularised risk minimisation applies to logistic regression. With an ℓ_1 penalty as in the lasso, we solve the minimisation problem

$$\min_{\boldsymbol{\beta}} \; -L(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) - \log \left(1 + \exp \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \right).$$

Multinomial logistic regression (key concept)

The **multinomial logistic regression** is a generalisation of logistic regression to multiple classes. The model specifies

$$p(y = c|\boldsymbol{x}) = \frac{\exp\left(\beta_{0c} + \boldsymbol{\beta}_c^T \boldsymbol{x}\right)}{\sum_{c'=1}^{C} \exp\left(\beta_{0c'} + \boldsymbol{\beta}_{c'}^T \boldsymbol{x}\right)},$$

where $\boldsymbol{\beta}_c$ is the vector of coefficients for class c.

The **softmax function**

$$\mathcal{S}_c(a_1, \ldots, a_C) = \frac{\exp\left(a_c\right)}{\sum_{c'=1}^{C} \exp\left(a_{c'}\right)}$$

ensures that the conditional class probabilities are in the $(0, 1)$ interval and add up to one.

Multinomial logistic regression

To avoid redundancy in the parameters, we usually specify $\beta_{0C} = 0$ and $\boldsymbol{\beta}_C = \mathbf{0}$. In this case,

$$p(y = c|\boldsymbol{x}) = \frac{\exp\left(\beta_{0c} + \boldsymbol{\beta}_c^T \boldsymbol{x}\right)}{1 + \sum_{c'=1}^{C-1} \exp\left(\beta_{0c'} + \boldsymbol{\beta}_{c'}^T \boldsymbol{x}\right)},$$

for $c = 1, \ldots, C-1$, and

$$p(y = C|\boldsymbol{x}) = \frac{1}{1 + \sum_{c=1}^{C-1} \exp\left(\beta_{0c} + \boldsymbol{\beta}_c^T \boldsymbol{x}\right)}.$$

The choice of baseline label does not affect the predictions. It is not necessary restrict the model in this way with regularised estimation.